Keskitetyn lokihallintajärjestelmän käyttöönotto

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Ammattikorkeakoulun opinnäytetyö

Tietojenkäsittelyn koulutusohjelma

Hämeenlinna, kevät 2017

Jani Hietanen

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Tietojenkäsittelyn koulutusohjelma
Visamäki, Hämeenlinna

Tekijä Jani Hietanen **Vuosi** 2017

Työn nimi Keskitetyn lokihallintajärjestelmän käyttöönotto

Työn ohjaaja/t Erkki Laine

TIIVISTELMÄ

Opinnäytetyön tavoitteena oli asentaa Logstash-, Elasticsearch- ja Kibana-ohjelmista koostuva lokihallintajärjestelmä, jonka tarkoituksena oli kerätä toimialueen virtuaalipalvelimilta sekä pfSense-reitittimeltä lokitietoa. Järjestelmän tarkoitus on vastaanottaa, prosessoida ja indeksoida lokitietoa niin, että se on selattavissa Kibana-verkkohallintapaneelin kautta. Käytännön työn loppuosassa tutustuttiin Kibanan visualisointityökaluihin, jonka tietolähteenä käytettiin toimialueelta kerättyjä lokeja. pfSensen-lokitiedoista luotiin hallintapaneeli, jonka avulla palomuurin tilaa voidaan seurata.

Työn teoriaosuudessa käsitellään lokeja sekä niiden hallinnasta koituvia hyötyjä ja ongelmia. Työssä syvennytään lokirakenteen ja sisällön lisäksi Windows- ja Linux-käyttöjärjestelmien lokilähteisiin sekä SNMP-lokiprotokollan toimintaan. Teoriaosuudessa tutustutaan myös käytännön osuudessa asennettaviin Elastic-ohjelmiin.

Työn tuloksena on toimiva lokihallintajärjestelmä, jota voidaan jatkokehittää eteenpäin.

Avainsanat loki, lokilähde, lokihallintajärjestelmä, elastic, pfsense

Sivut 43 sivua, joista liitteitä 5 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Degree Programme in Business Information Technology
Visamäki, Hämeenlinna

Author Jani Hietanen **Year** 2017

Subject Installation of a centralized logging system

Supervisors Erkki Laine

ABSTRACT

The goal of this thesis was to install a centralized logging system that would consist of Logstash, Elasticseach and Kibana. The purpose of this system was to gather logging information from virtualized servers and pfSense router in the domain. Logging data would then be processed and indexed so that it could be browsed from Kibana that would act as a web-based user interface. In the last practice section Kibana is being used as a visualization tool for the gathered log data. Log data from pfSense was then applied to create a dashboard that could be used for monitoring the firewall.

The theory section of this thesis consists of information about benefits and problems of logging and log management. It also provides an in-depth look into the structure and contents of logs and talks about Windows- and Linux-operating systems as logging sources and SNMP-protocol. It also introduces Logstash, Kibana, Elasticsearch and Beat-programs that will be then installed in the practice section.

The end result is a functional centralized logging system that can be developed further.

Keywords log, log source, centralized logging system, elastic, pfsense

Pages 43 pages including appendices 5 pages

SANASTO

NIMI	SELITYS
grok	Logstash-suodatin, jolla voidaan jäsentää tietoa tietojanasta ja asettaa sille merkitys (Elasticsearch n.d.).
indeksi	Tiedon hakua nopeuttava optimointi.
ip-osoite	Numerotunniste, jolla tietokone tai laite voidaan paikantaa TCP/IP-verkossa.
klusteri	Joukko tietokoneita, jotka on yhdistetty toisiinsa muodostamaan kokonaisuuden, jossa yhdistyy kaikkien yhteen liitettyjen tietokoneiden laiteteho.
loki	Tietokoneohjelman luoma teksti, joka kertoo ohjelman tai järjestelmän tapahtumasta.
lokitus	Termi, jolla tarkoitetaan lokien luontia.
ohjelmistopino	Koostuu useasta eri ohjelmasta, jotka yhdessä muodostavat ohjelmistokokonaisuuden tietyn tavoitteen saavuttamiseksi. Esimerkkinä LAMP-ohjelmistopino, joka koostuu useasta ohjelmasta. Pinon ohjelmat muodostavat yhdessä verkkoalustan, jolla voidaan luoda ja hallinnoida verkkosivuja.
paketinhallintaohjelma	Paketinhallintaohjelma on ohjelma, jolla hallitaan Linux- & Unix-käyttöjärjestelmien ohjelmia. Hallintaohjelmalla voidaan ladata ohjelmia pakettivarastosta sekä päivittää tai poistaa niitä tietokoneelta.
pakettivarasto	Pakettivarasto (engl. repository) on palvelimelle keskitetty ohjelmistokirjasto, josta ohjelmia voidaan ladata internetin kautta tietokoneelle.
protokolla	Protokolla on tiedonsiirron standardi, joka koostuu määrätyistä säännöistä ja ohjeistuksista.
SSL-suojaus	SSL (Secure Sockets Layer) on sovellustason protokolla, jonka tarkoitus on suojata tietokoneiden välinen tiedonsiirto (Ubuntu n.d.).
tietokanta	Organisoitu kokoelma tietoja, jonka tarkoitus on tehostaa tiedon hakua.
virtuaalikone	Virtualisoinnin avulla luotu koneympäristö, joka toimii normaalin tietokoneen tavoin.
virtualisointialusta	Alusta, jolla voidaan luoda virtuaalikoneita. Sen tehtävä on hallinnoida ja jakaa isäntäkoneen laitteistoresursseja virtuaalikoneille. Esimerkkejä virtualisointialustoista ovat VMware vSphere (ESXi), Microsoft Hyper-V ja Citrix XenServer.
visualisointi	Graafinen esitys, joka koostuu kerätystä datasta.

SISÄLLYS

1 JOHDANTO 1

2 LOKIT JA NIIDEN KÄYTTÖ 2

2.1 Loki 2
2.1.1 Rakenne 2
2.1.2 Sisältö 3
2.1.3 Hyödyt 4
2.1.4 Ongelmat 6
2.2 Lokilähteet 8
2.2.1 Syslog-protokolla 8
2.2.2 SNMP-protokolla 10
2.2.3 Windows Event Log 13
2.3 Lokien varastointi 16
2.4 Lokivalvonnan tasot 17
2.5 Lokitietojen kerääminen 18

3 KÄYTÄNNÖN OSUUDEN ESIVALMISTELU 20

3.1 Tietoa Elastic-ohjelmista 20
3.1.1 Elasticsearch-hakumoottori 20
3.1.2 Logstash-lokienkäsittelyohjelma ja Kibana-käyttöliittymä 21
3.1.3 Filebeat- ja Winlogbeat-lokienlähetysohjelmat 22
3.2 Suunnitelma 23
3.3 Toteutusympäristö 24

4 ELASTIC-LOKIHALLINTAJÄRJESTELMÄN KÄYTTÖÖNOTTO 26

4.1 Elastic-ohjelmien asennus 26
4.2 Elastic-palvelimen konfigurointi 27
4.2.1 Elasticsearchin indeksointipohja ja asetukset 27
4.2.2 SSL-sertifikaatin ja avaimen luonti 28
4.2.3 Logstashin konfigurointi 29
4.2.4 Kibanan konfigurointi 33
4.3 Asiakaskoneiden konfigurointi 34
4.3.1 Lokien lähetys Linux-palvelimilta ja pfSense-reitittimeltä 34
4.3.2 Lokien lähetys Windows-palvelimilta 36
4.4 Asennuksen jälkeinen testaus 37
4.5 Tiedon visualisointi 40

5 YHTEENVETO 43

LÄHDELUETTELO 44

Liitteet

Liite 1 pfSensen grok-kaavat

Liite 2 Logstashin pfSense-suodatin

Liite 3 pfSensen palomuurilokitiedoista koostuva hallintapaneeli

1 JOHDANTO

Ennen opintojeni alkua kiinnostuin palvelimista ja niiden hallinnasta. Huomasin nopeasti, kuinka aiheesta tuli myös vapaa-ajan harrastus. Kotiverkossani olikin pian virtualisointipalvelin, jota olen oppinut hallitsemaan. Halusin kehittää virtuaalipalvelimieni hallintaa ja kerätä enemmän tietoa niiden toiminnasta. Huomasin kiinnostuvani yhä enemmän lokeista ja niiden tuomista hyödyistä järjestelmien valvonnassa. Kiinnostuksen innoittamana valitsin tämän opinnäytetyön aiheen.

Ihmisten odotusarvot verkkopohjaisia palveluita kohtaan ovat kasvaneet niiden yleistymisen myötä. Verkkoratkaisut on luotu helpottamaan ihmisten elämää, ja yhä useampi ihmisistä haluaa mieluummin hoitaa asiointinsa verkossa kuin paikan päällä. Näiden palveluiden sekä muiden perinteisten yritysten IT-infrastruktuuri perustuu palvelimien ylläpitämiseen. Yksi syy digitalisaation kovaan vauhtiin on virtualisoinnin ja pilvipalveluiden mahdollistama kustannustehokkuus, joka takaa yrityksille entistä edullisemmin palvelintehoa palveluiden tuottamiseen. Yrityksellä voi olla tänä päivänä hallittavinaan satoja tai jopa tuhansia työasemia ja palvelimia useissa eri toimipisteissä ympäri maailmaa. Palvelimien ylläpidosta vastaavat tietohallinnot ja IT-tukiryhmät ovat aivan uuden tason haasteiden edessä ja niiden toiminnan kannalta on tärkeää saada jatkuvaa tietoa koneiden ja ylläpidettävien järjestelmien tilasta.

Markkinoilla on monia valvontaohjelmia, jotka osaavat kertoa tietokoneiden tilasta, mutta järjestelmiä koskevan yksityiskohtaisen tiedon saamiseen on käytettävä lokeja. Lokit voivat paljastaa syvemmän tason ongelmia, jotka eivät muuten ole valvontaohjelmien nähtävissä. Lokien analysointi ja tulkinta voi olla haastavaa. Opinnäytetyössä tutustutaan lokeihin ja selvitetään, miten niitä voidaan hyödyntää, samalla läpikäyden niihin liittyviä ongelmakohtia. Käytännön työn esivalmisteluosassa käydään läpi Elastic-lokikäsittelyohjelmia, joihin kuuluvat Logstash, Elasticsearch, Kibana ja lisäohjelmina toimivat Beatsit. Ohjelmat asennetaan käytännön osuudessa. Tavoitteena on luoda niiden avulla keskitetty lokihallintajärjestelmä. Lopuksi testataan Kibanan visualisointityökaluja hyödyntämällä kerättyä lokitietoa. Ohjelmat valittiin, koska ne perustuvat avoimeen lähdekoodiin, eivätkä näin ollen aiheuta ylimääräisiä kustannuksia. Tämän lisäksi Elastic-ohjelmista oli hyvin saatavilla asennusoppaita ja ohjeita. Projekti toteutetaan kodin lähiverkossa.

Työssä vastataan tutkimuskysymyksiin:
Mitä ovat lokit?
Miten lokeja voidaan hyödyntää?
Mitkä ovat Windows- ja Linux-järjestelmien lokien eroavaisuudet?
Mitä hyötyä lokienkeskittämisestä on?

2 LOKIT JA NIIDEN KÄYTTÖ

2.1 Loki

Lokit kertovat järjestelmän tai sovelluksen ajon aikaisista tapahtumista. Niihin tallentuu tietoa siitä, miten järjestelmä tai ohjelma on reagoinut eri tilanteisiin. Lokeja syntyy laite- ja sovellustasolla. Esimerkiksi ohjelmat, verkkolaitteet ja tietokoneiden käyttöjärjestelmät voivat generoida omasta toiminnastaan lokeja. (Alapati 2016.) Tässä luvussa käydään läpi mitä lokit ovat, mitä ne sisältävät ja miten niitä voidaan hyödyntää järjestelmien valvonnassa. Luvussa tarkastellaan myös lokihallinnan ongelmakohtia sekä syvennytään kolmeen suureen lokilähdetyyppiin. Loppuosassa käydään läpi lokien varastointitapoja ja paneudutaan lokienvalvontaan sekä lokitiedon keräämiseen.

2.1.1 Rakenne

Lokien rakenteelle ei ole olemassa yhtenäistä standardia. Tästä johtuen niiden rakenne voi vaihdella käyttöjärjestelmästä tai ajettavasta ohjelmasta riippuen. Lokilähteiden tallennusformaatteja on useita, esimerkkejä niistä ovat syslog, csv, Simple Network Management Protocol (SNMP), Extensible Markup Language (XML) ja binääritiedostot. Tämän lisäksi kolmannen osapuolen ohjelmistot voivat käyttää omia sovelluskohtaisia formaatteja. (Kent & Souppaya 2006, 23.)

Vaikka lokiviesteissä voi olla eri järjestelmistä ja protokollista johtuvia rakenteellisia eroja, sisältävät ne yleensä kolme perustietoa, jotka ovat aikaleima, lähde ja viestisisältö (Chuvakin, Schmidt & Phillips 2012, 6). Aikaleimasta käy ilmi ajankohta, jolloin lokiviesti on luotu. Lokiviestin lähde kertoo mistä järjestelmästä viesti on peräisin. Lähde voi olla joko tietokoneen IP-osoite tai isäntänimi. Viestisisältö sisältää lokiviestin viestiosuuden.

Alapati (2016) vertaili Tomcat ja Apache verkkopalvelimien lokiviestejä esitellessään lokiformaattien eroavaisuuksia. Kuvasta 1 Alapati nosti esiin aikaleimojen eri formaatit, jotka voivat hankaloittaa samanaikaisten tapahtumien korrelointia. Lokeja verratessa nähdään myös, kuinka tiedon asettelutavat eroavat toisistaan. Tomcat-verkkopalvelimen lokiviesti alkaa päiväyksellä, toisin kuin Apachen loki, joka alkaa viestin lähteellä, eli tietokoneen IP-osoitteella.

```
#Tomcat Web server log entry
Jun 24, 2016 4:58:32 PM org.apache.catalina.startup.HostConfig deployWAR
INFO: Deployment of web application archive \soft\apache-tomcat-7.0.62\webapps\samp
has finished in 264 ms

#Apache log entry
127.0.0.1 - - [24/Jun/2016:16:54:58 +0530] "GET /favicon.ico HTTP/1.1" 200 21830
```

Kuva 1. Tomcat- ja Apache-verkkopalvelimien lokivertailu (Alapati 2016).

2.1.2 Sisältö

Lokien tärkein osuus on viestin sisältö. Lokiviestin sisällöstä voidaan vetää johtopäätöksiä järjestelmien, ohjelmien ja laitteiden toiminnasta, sekä laitteistoresurssien käytöstä. Lokit voivat paljastaa viesteillään viallisia ja viallisiksi meneviä laitteistokomponentteja. Haluttaessa ne mahdollistavat myös käyttäjien seurannan ja voivat näin paljastaa, mitä käyttäjä on työasemallaan tehnyt. Lokiviestit voivat sisältää tietoa tietoturvamurroista, niiden yrityksistä ja kertoa kuka vieraili milläkin verkkosivulla. Sovellustasolla ne kertovat ohjelmien toiminnasta ja niiden aiheuttamista muutoksista. (Chuvakin, Schmidt & Phillips 2012, 45.)

Viestintävirasto (2016, 2–3) luokittelee lokit useaan eri tyyppiin:

- ylläpitoloki
- tapahtumaloki
- muutosloki
- virheloki
- pääsynvalvontaloki
- viestintäloki
- haltijaloki.

Ylläpitolokiin tallentuu tieto järjestelmien ja käyttöoikeuksien muutoksista. Tapahtumaloki kirjaa käyttäjäkirjautumiset sekä järjestelmäprosessien tekemät muutokset ja tapahtumat. Muutosloki paljastaa käyttäjien tekemät muutokset tietoihin. Virhelokia käytetään vikatilanteiden selvitykseen ja niiden kirjaamiseen. Viestintälokin avulla esimerkiksi sähköpostipalvelimella voidaan jäljittää mistä sähköposti on lähetetty, kenelle se on osoitettu sekä milloin se on lähetetty. Pääsynvalvontalokiin kirjataan käyttäjien kirjautumiset ja sitä voidaan käyttää tietoturvan valvontaan. Haltijalokilla voidaan jäljittää kenelle, jokin tietty verkko- tai domain-osoite kuuluu. (Viestintävirasto 2016, 2–3.)

2.1.3 Hyödyt

Lokeista on huomattava hyöty järjestelmien valvonnassa sekä vianselvitystyössä. Verkkolaitteet kuten reitittimet, kytkimet, palomuurit ja tukiasemat pystyvät yleensä generoimaan toiminnastaan lokeja. Palomuurit ja reitittimet kirjaavat tietoa läpikulkeneesta tai estetystä liikenteestä lokeihin. Lokeja seuraamalla voidaan nähdä tietokoneen verkkoliikenne ja sen eteneminen verkossa.

Lokitietojen hyödyntämiseen vaaditaan tuntemusta valvottavasta ympäristöstä sekä ylläpidettävistä järjestelmistä ja ohjelmista. Virhelokit eivät aina osoita suoraan ongelman lähteeseen, vaan niitä on ajateltava syiden ja seurausten avulla. Esimerkiksi virhelokiin ilmestynyt viesti epäonnistuneesta VPN-yhteydenotosta ei paljasta missä varsinainen ongelman lähde on vaan kertoo sen, että VPN-palvelin ei syystä tai toisesta vastannut yhteydenottopyyntöön, joka on johtanut yhteyden epäonnistumiseen. Ratkaisu ongelmaan voi olla yksinkertainen. VPN-palvelin voi olla kaatunut tai sammunut, eikä näin kykene vastaamaan tietokoneen yhteydenottoihin. Vastaus voi olla myös hankalampi. VPN-palvelimen julkinen IP-osoite on voinut vaihtua tai palvelimen palomuuri on mahdollisesti estänyt yhteyden muodostumisen. Nämä kaikki voivat selittää virhelokiin syntyneen viestin. Järjestelmänvalvojan on seurattava lokien jättämiä johtolankoja ja yksi kerrallaan poissuljettava eri mahdollisuuksia, kunnes oikea ratkaisu vikaan löytyy.

Viestintäviraston (2016, 3) lokiohjeessa käydään läpi lokeja ja korostetaan niiden tärkeyttä tietoturvan seurannassa. Lokit kirjaavat tarkasti järjestelmän eri osa-alueiden tapahtumia, joista yhdessä saadaan kattava kokonaiskuvan. Viestintälokilla voidaan seurata tietoliikenteen kulkua lähiverkossa, kun taas järjestelmään kohdistunut murtautuminen jättää jälkiä pääsynvalvontalokiin. Muutoslokista voidaan nähdä mitä tiedostoja on muokattu, jolloin tiedetään mitä murtautuja on järjestelmässä tehnyt. Tietoturvaohjelmistot, palomuurit sekä muut verkkolaitteet tuottavat lokeja, joita voidaan käyttää tietoturvan valvontaan sekä tietoturvamurron jälkeisissä tutkimuksissa (Messier 2015, 199).

Käyttäjien aktiviteeteista tallentuu jatkuvasti tietoa lokeihin, joista voi olla apua yritysten tai organisaatioiden murron jälkeisissä tutkimuksissa. Aikaleimattujen lokien avulla järjestelmän tapahtumista voidaan luoda aikajana. Vaihtoehtoisesti Windows-käyttöjärjestelmissä voidaan ottaa käyttöön käyttäjien yksityiskohtaisempi lokitus, jonka avulla käyttäjän toimista jää jälki tietokoneen lokijärjestelmään. Tämän avulla voidaan selvittää, mitä käyttäjä on työasemallaan tehnyt ja mihin tiedostoihin hän on koskenut. Sen avulla selviää myös, jos käyttäjä on yrittänyt avata oikeuksilleen sopimatonta materiaalia. (Messier 2015, 199.)

Virustorjuntaohjelman lokit ovat hyödyllisiä, sillä ne voivat kertoa, jos järjestelmä on saastunut tai vaarantunut. Chuvakin, Schmidt ja Phillips (2012, 25–26) nostavat esiin esimerkkitapauksen, jossa tietokonetta on tutkittu järjestelmän hitauden takia. Virustorjuntaohjelman lokeista nähtiin, että ohjelman automaattiset tietoturvapäivitykset olivat epäonnistuneet kovalevyn tilanpuutoksen vuoksi, joka oli tehnyt järjestelmästä haavoittuvan. Lokit paljastivat kaksi haittaohjelmaa, jotka virustorjuntaohjelma oli asettanut karanteeniin sekä kolmannen haittaohjelman, jota virustorjuntaohjelma ei onnistunut eristämään (Kuva 2). Virustorjuntaohjelman lokien ansiosta tiedettiin ajankohta, milloin ensimmäiset tartunnan merkit ilmenivät sekä haittaohjelmien tiedostonimet ja sijainnit järjestelmässä. Lokit kertoivat kovalevytilan puutoksesta ja epäonnistuneista päivityksistä, joista muodostui riskitekijä.

```
Virus Found!Virus name: W32.Welchia.B.Worm in File: C:\WINNT\
   system32\drivers\svchost.exe by: Manual scan. Action: Quarantine
   succeeded:
Virus Found!Virus name: W32.Randex.gen in File: C:\WINNT\system32\
   wumgrd.exe by: Manual scan. Action: Quarantine succeeded:
Virus Found!Virus name: Backdoor.IRC.Bot in File: C:\WINNT\system32\
   mfm\msrll.exe by: Manual scan. Action: Clean failed: Quarantine
   failed:
```

Kuva 2. Virustorjuntaohjelman lokit (Chuvakin, Schmidt & Phillips 2012, 26).

Virustorjuntaohjelmien lisäksi tietokoneiden käyttöjärjestelmät kertovat järjestelmän toiminnan kannalta tärkeistä tapahtumista, kuten palveluiden, ohjelmien ja laitteiston toiminnasta. Tästä johtuen lokitusta käytetäänkin yleisimmillään vianselvitystyöhön. Eri laite- ja ohjelmistoviat voivat ilmetä lokeista jo varhaisessa vaiheessa, jolloin valvonnan alainen järjestelmä voidaan huoltaa nopeasti, säästäen vianetsimiseen kuluvaa aikaa. (Chuvakin, Schmidt & Phillips 2012, 18.)

Lokien analysoinnista on myös apua järjestelmän suorituskyvyn optimoinnissa ja pullonkaulojen etsinnässä. Jotta järjestelmä voidaan optimoida, on järjestelmänvalvojalla oltava tarpeeksi tietoa asennetuista ohjelmista, niiden laitteistovaatimuksista sekä järjestelmän laiteresurssien käytöstä. Lokien aikaleimojen avulla ylläpidettävän järjestelmän toiminnasta voidaan luoda aikajana, jonka avulla järjestelmän käyttäytymistä ja toimintaa voidaan seurata pidemmältä aikaväliltä. Palvelimella ajettavien verkkopalveluiden toimintaa ja kuormitusta voidaan seurata esimerkiksi vasteaikojen avulla. (Chhajed 2015, 2.)

Lokiviestien yhteydessä käytetään usein erilaisia työkaluja lokitiedon hakemiseen. Linux-pohjaisessa käyttöjärjestelmässä voidaan käyttää grep-työkalua, jonka avulla lokitiedoston sisältämien lokien joukosta voidaan hakea avainsanalla tiettyä tapahtumaa tai vikaviestiä. Tällaisia työkaluja käyttäessä järjestelmänvalvojan on tiedettävä millaisia viestejä etsiä, jotta hän voi löytää oikeat lokiviestit. (Chhajed 2015, 2.)

Verkkosivujen käyttäjämääriä voidaan seurata verkkopalvelimen lokien avulla. Ne voivat kertoa mitä käyttäjä on verkkosivulla tehnyt ja mitä kuvia tai tiedostoja käyttäjä on sivuilla ladannut tai katsonut. Uutissivustot voivat tällä tavoin seurata artikkeleiden suosiota ja saada sen avulla tietoa minkälaisista kirjoituksista heidän lukijansa pitävät.

2.1.4 Ongelmat

Yleinen ongelma lokien tarkastuksessa ja keräyksessä on niiden vaatimat resurssit. Riippuen hallittavien tietokoneiden ja laitteiden määrästä lokien tehokkaaseen valvontaan voidaan joutua panostamaan huomattavasti laiteresursseja, kuten prosessoritehoa, keskusmuistia, levykapasiteettia sekä verkkokaistaa. Tästä syystä yritys voi tehdä tietoisen päätöksen jättää lokivalvonta kokonaan hyödyntämättä ja keskittää olemassa olevat resurssit muiden palveluiden ylläpitämiseen. (Verizon 2015, 59.)

Lokiseurannasta tulee vaikeampaa mitä enemmän hallittavia työasemien ja palvelimia ympäristössä on. Windows-palvelin voi generoida jopa yli 5 000 lokitapahtumaa päivässä (Covington 2015). Suurin osa tuosta tiedosta on järjestelmän normaalia toimintaa koskevaa tietoa, jolloin hyödyllisen tiedon löytämisestä muodostuu huomattava haaste. Tämän takia toimialueen kasvaessa myös valvontaa on kehitettävä vastaamaan paremmin kasvavan ympäristön haasteita. Lokien keskittäminen on askel oikeaan suuntaan, sillä se tarjoaa selvän ratkaisumallin, jolla pystytään paremmin hallitsemaan lokeja sekä auttamaan merkityksellisen tiedon löytämistä ja keräämistä. (Turnbull 2016, 6.) Lokien keskittämisellä tarkoitetaan lokitiedon ohjausta eri tietokoneilta ja laitteilta keskitetylle lokihallintapalvelimelle.

Lokien keskitys ei tässä vaiheessa vielä riitä kun kaikki lokit ohjataan suoraan keskitetylle palvelimelle, muodostuu siitä valtava määrä tietoa, jonka läpikäyminen on hyvin vaikeaa (Turnbull 2016, 7). Avaintekijä lokihallintajärjestelmän käyttöönotossa on huolellinen suunnittelu. Ennen lokien keräystä on tiedettävä, minkälaista lokitietoa halutaan kerätä ja mihin tarkoitukseen. Lokihallintajärjestelmä on suunniteltava niin, että sillä kerätään vain ennalta sovittua ja järjestelmän hallinnan ja valvonnan kannalta oleellista tietoa. Näin saapuvan lokitiedon määrää saadaan vähennettyä. (Viestintävirasto 2016, 6.)

Toinen ongelma lokien hyödyntämisessä ja keräämisessä on niiden eri lähteet ja formaatit. Etenkin yritysten tietoverkoissa on normaalia nähdä eri käyttöjärjestelmillä toimivia työasemia ja palvelimia. Windows- ja Linux-käyttöjärjestelmät generoivat lokeja eri formaatteihin. Tämän lisäksi kolmannen osapuolen ohjelmat ja palvelut käyttävät usein omia lokiformaatteja ja muotoja. Yhtenäisen lokistandardin puuttuminen on seuraava haaste lokien keskittämistä suunnittelevalle organisaatiolle.

Lokistandardin puuttumisesta johtuen lokeissa oleva tieto, aikaleima ja lähde voivat olla viestissä eri järjestyksessä, päivämäärät eri formaateissa sekä viestin sisältö voivat olla eri lailla raportoitua lähteestä riippuen. Tämä vaikeuttaa lokien tulkintaa ja läpikäyntiä. Lokiviestiin merkitty tietoliikenneportti voi olla numero, kun taas toisen ohjelman lokissa saman porttinumeron tilalla voi olla kirjainlyhenne. Esimerkiksi lokiviestissä olevaan Secure Shellin porttiin voidaan viitata joko numerolla 22 tai kirjaintunnuksella SSH. Molemmat tarkoittavat samaa asiaa, mutta järjestelmänvalvojan on myös tämä tiedostettava, jotta hän voi ratkaista tämän tason ongelmia. Lokihallintajärjestelmässä eri formaatissa ja muodossa olevat lokit muutetaan määriteltyyn standardimuotoon. (Kent & Souppaya 2006, 22–23.)

Lokiformaateista johtuvien ongelmien lisäksi myös järjestelmän aiheuttama väärä aikaleima voi tuottaa ongelmia lokien analysoinnissa. Lokien aikaleimat muodostuvat isäntäkoneen kellonajan mukaisesti, eli jos tietokoneen kellonaika on väärä, vääristää se samalla myös lokien aikaleiman. Tämä tekee lokien analysoinnista vaikeaa etenkin, jos verrattavissa ovat kahden tai useamman tietokoneen lokit. Järjestelmien tapahtumia ja reaktioita voi olla vaikea yhdistää toisiinsa, jos tapahtumat eivät ajallisesti korreloi keskenään. (Kent & Souppaya 2006, 23.)

Lokeista saatu tieto voi olla hyvinkin hyödyllistä sovelluksista johtuvien vikojen korjaamisessa. Ongelmana kuitenkin on se, että lokit kertovat vain sen mitä ohjelman kehittäjät ovat ohjelman lokijärjestelmään ohjelmoineet. Jotta lokeista olisi hyötyä vianselvityksessä, sovelluksen kehitysvaiheessa sovelluskehittäjän on pitänyt osata varautua tietyn ongelman mahdollisuuteen ja ohjelmoida sovelluksen lokijärjestelmä kertomaan tarpeeksi viasta ja sen synnystä, jotta järjestelmänvalvoja voisi päätellä vikaviestistä mahdollisia korjaustoimia. Jotkut ohjelmat voivat viestiä virheiden lisäksi hyvin yksityiskohtaisesti ohjelman toiminnasta, kun taas toiset ohjelmat kirjaavat lokeihin vain kriittiset viat. (Grimes 2012, 3.)

2.2 **Lokilähteet**

Lokilähteillä tarkoitetaan protokollaa, jolla lokit luodaan. Lokienkirjausjärjestelmä on osa käyttöjärjestelmää, ohjelmaa tai laitteiston firmware-ohjelmistoa. Windows-käyttöjärjestelmien lokikirjausjärjestelmää kutsutaan tapahtumienvalvonnaksi (engl. Event Log). Linux- ja Unix-järjestelmissä käytetään Syslog-protokollaa lokien luomiseen ja verkkolaitteissa voidaan käyttää SNMP-protokollaa. Opinnäytetyön aiemmassa osassa (Luku 2.1.1) puhuttiin lokien eri formaateista ja rakenteista. Formaattien erot johtuvat erityyppisten lokilähteiden käytöstä, joihin tässä luvussa syvennytään.

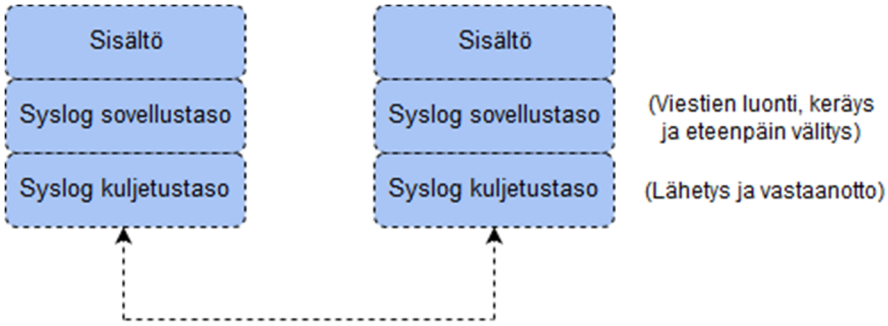
Lokilähteitä on kahdentyyppisiä. Lokilähteet joko lähettävät lokitietoa (engl. push) tai hakevat (engl. pull) sitä. Push-pohjaisissa lokilähteissä lokiviesti viedään joko paikalliselle kiintolevylle tai verkossa olevalle lokien kerääjälle (engl. log collector). Käytetyimpiä push-pohjaisia lokilähteitä ovat Syslog, SNMP ja Windows Event Log. Ohjelma voi myös hakea lokiviestin suoraan lokilähteestä, jolloin lokilähde on pull-pohjainen. Lokitieto voidaan tällöin tallentaa tarvittaessa suoraan tietokantaan. (Chuvakin, Schmidt & Phillips 2012, 51–52.)

2.2.1 Syslog-protokolla

Syslog-protokollaa käytetään Unix-käyttöjärjestelmissä kernelin, eli järjestelmän ytimen, ja sovellusten lokitietojen generoimiseen. Syslog voidaan konfiguroida vastaanottamaan lokitietoja myös muilta tietokoneilta. (Chuvakin, Schmidt & Phillips 2012, 52.) Tämän ominaisuuden avulla Syslog-palvelimesta voidaan rakentaa lokihallintajärjestelmä (Messier 2015, 205).

Syslog koostuu kolmesta tasosta, jotka ovat sisältö-, sovellus- ja kuljetustasot. Syslog-lokiviestin tieto sijaitsee sisältötasolla. Sovellustaso hoitaa lokien luonnin, tulkinnan, reitityksen, sekä lokiviestien varastoinnin. Kuljetustason tehtäväksi jää lähettää ja vastaanottaa lokiviestejä verkon kautta. (RFC 5424/2009, 4.)

Kuvasta 3 käyvät ilmi Syslogin käsitteelliset tasot. Sovellustasolla luodaan Syslog-viestin sisältö. Syslog-viestin sisältö kerätään sovellustasolla analysoitavaksi. Sovellustasolla myös välitetään ja hyväksytään viestejä muilta lähettäjiltä. Kuljetustasolla Syslog-viesti viedään määrätylle lähetysprotokollalle, kun taas toisessa päässä lähetyksen vastaanottaja vastaanottaa Syslog-viestit lähetysprotokollalta. (RFC 5424/2009, 5.)



Kuva 3. Syslogin toiminnan tasot (RFC 5424/2009, 5).

Syslogilla on useita heikkouksia. Yksi niistä on UDP-protokollan käyttäminen verkon yli lähetetyissä viesteissä. Protokollasta johtuen lähettäjä ei saa vahvistusta lähetyksen vastaanottamisesta. (Alapati 2016.) Verkon yli lähetettyjä lokiviestejä on mahdollista kaapata käyttämällä verkon analysointiin erikoistuvia työkaluja. Syslog ei myöskään suojaa viestejä, joten niiden sisältö on kaappaajan luettavissa. Verkkoon murtautunut hyökkääjä voi saada lokien avulla lisätietoa verkon laitteista, järjestelmistä ja palveluista, jota hän voi hyödyntää etsiessään uusia heikkouksia. Edellä mainitut haavoittuvuudet, johtavat myös siihen, että Syslog-viestejä voidaan väärentää esimerkiksi netcat-työkalun avulla. (Kenneth 2003, 5.) Netcat on työkalu, jolla voidaan muun muassa kuunnella TCP- & UDP-portteja ja lähettää omia UDP-paketteja verkon yli. (Oracle, 2016.)

Muihin työkaluihin yhdistettynä väärennettyjä viestejä voidaan käyttää lähiverkkoliikenteen saturointiin, jolloin verkon sisäistä liikennettä voidaan kuormittaa Denial of Service (DoS) -hyökkäyksen avulla. (Kenneth 2003, 5.) Tämän toteuttaminen vaatisi hyökkääjältä kuitenkin pääsyä kohteen lähiverkkoon, eli tietoturvatoimet olisi jo tässä vaiheessa kertaalleen pahasti pettäneet.

Syslogin pohjalta on luotu avoimeen lähdekoodiin perustuva, monin tavoin paranneltu ohjelma nimeltä Syslog-ng, joka korjaa monia alkuperäisen Syslogin heikkouksia. Syslog-ng käyttää UDP-protokollan sijasta TCP-protokollaa, jolla voidaan varmistaa, että lähetetyt viestit ovat saapuneet perille. Tämän lisäksi ohjelma mahdollistaa lokiviestien salauksen TLS-protokollan avulla. (Balabit 2017.)

2.2.2 SNMP-protokolla

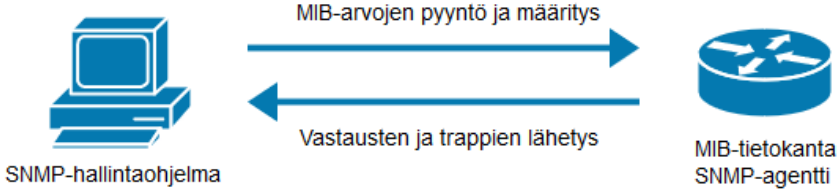
SNMP (Simple Network Management Protocol) on vuonna 1988 kehitetty protokolla verkkolaitteiden valvontaan. SNMP-protokolla mahdollistaa myös verkkolaitteiden hallinnan etänä. Protokollan avulla järjestelmänvalvoja voi saada laitteiden toiminnasta tärkeää tietoa, kuten vikaviestejä tai laitetietoja. SNMP-protokollaa voidaan käyttää myös laitevikojen paikantamiseen. Protokollasta on tällä hetkellä kolme versiota: SNMPv1, SNMPv2 ja SNMPv3. (Mauro & Schmidt 2005, 1–2.)

SNMP ei sinänsä ole lokijärjestelmä, mutta järjestelmien sekä laitteiston tilasta kertovat trapit ja ilmoitukset toimivat lokiviestien tavoin. Monet verkkolaitteet kykenevät lähettämään lokiviestejä myös Syslogin kautta, mutta kaikki vanhat verkkolaitteet eivät tähän kykene. Tällöin lokitietojen keräämiseen voidaan käyttää SNMP-protokollaa. (Chuvakin, Schmidt & Phillips 2012, 59.)

SNMP-protokolla toimii agenttien, hallintaohjelman sekä MIB-tietokannan avulla. SNMP-agentti on joko erillinen ohjelma tai verkkolaitteen järjestelmään integroitu osa, joka kerää tietoa laitteesta tietokantaan, jota kutsutaan lyhenteellä MIB (Management Information Base). Järjestelmänvalvoja voi pyytää SNMP-hallintaohjelman kautta tietoa laitteesta ja sen tilasta get-komennon avulla, jolloin SNMP-agentti hakee tiedon MIB-tietokannasta ja välittää sen takaisin hallintaohjelmalle. Hallintaohjelmalla voidaan myös määrittää tietoa tietokantaa set-komennon avulla. SNMP-hallintaohjelmaa voidaan kutsua myös lyhenteellä NMS (Network Management Station). (Mauro & Schmidt 2005, 3–4.)

Nykyajan verkkolaitteissa on usein sisäänrakennettu SNMP-agentti. Vian tai ongelman sattuessa SNMP-agentti lähettää hälytysviestin SNMP-hallintaohjelmalle. Tällaista viestiä kutsutaan nimellä trap. Viestin saatuaan hallintaohjelma voi ohjeistuksestaan riippuen lähettää hälytysviestin järjestelmänvalvojalle. Trap-viestit ovat SNMP-agentin lähettämiä itsenäisiä viestejä. (Mauro & Schmidt 2005, 3–4.)

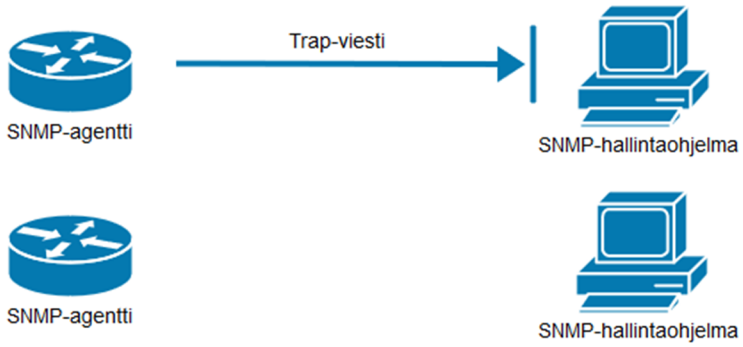
Alla oleva kuva (Kuva 4) esittää SNMP-hallintaohjelman ja SNMP-agentin välistä tiedonvälitystä. SNMP-hallintaohjelma voi esittää kyselyjä ja määrityksiä SNMP-agentin välityksellä MIB-tietokannalle. SNMP-agentti voi vastata pyyntöihin ja lähettää itsenäisesti trap-viestejä hallintaohjelmalle.



Kuva 4. SNMP-hallintaohjelman ja agentin välinen viestintä (Cisco n.d.).

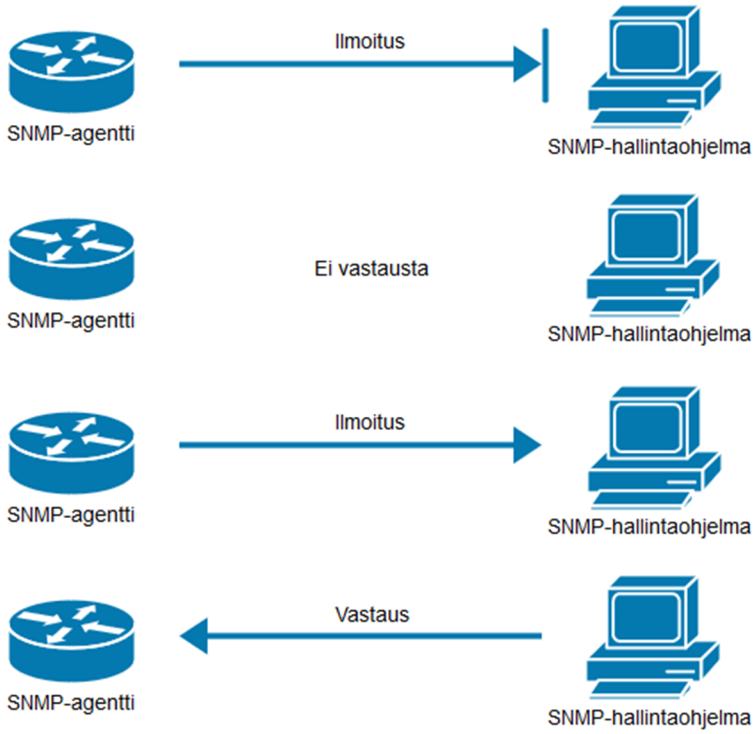
SNMP käyttää Syslogin tavoin UDP-protokollaa viestittelyyn, mutta toisin kuin Syslog, SNMP-hallintaohjelma valvoo ohjelmasta lähetettyjä UDP-paketteja, jos SNMP-agentti ei vastaa hallintaohjelman pyyntöön tietyssä ajassa, olettaa se paketin kadonneen ja uudelleen lähettää sen. UDP-protokollan hyöty tuleekin siitä, että sen käyttäminen kuormittaa verkkoa vähemmän kuin TCP-protokollan käyttäminen, jossa jokaisen paketin kohdalla tarkistetaan, että se on saapunut perille. (Mauro & Schmidt 2005, 19.)

SNMP-protokollalla on myös omat heikkoudet. SNMP:n ongelmaksi koituvat SNMP-agentin lähettämät trap-viestit. Toisin kuin SNMP-hallintaohjelman lähettämät kyselyt, joiden perille saapuminen tarkistetaan, trap-viestien lähetyksistä hallintaohjelma ei ole tietoinen, ennen kuin viesti saapuu ohjelmaan. Trap-viestien lähetyksessä käytetään UDP-protokollaa, jolloin SNMP-agentti ei ole tietoinen viestin perille pääsystä. Tästä johtuen SNMP-agentti ei kykene uudelleenlähettämään trap-viestiä lähetyksen epäonnistuessa (Kuva 5). (Mauro & Schmidt 2005, 19.)



Kuva 5. SNMP-agentti ei tarkista trap-viestien perille saapumista (Cisco n.d.).

SNMPv2- ja SNMPv3-protokollissa trapien sijasta on mahdollista käyttää ilmoituksia. Ilmoitukset eroavat trapeista siten, että ilmoituksen vastaanottajan on kuitattava SNMP-agentin lähettämä ilmoitus vastauksella, jos SNMP-agentti ei saa lähetetystä viestistä vastausta hallintaohjelmalta, uudelleen lähettää se viestin tietyn ajan kuluessa (Kuva 6). (Chuvakin, Schmidt & Phillips 2012, 60.)



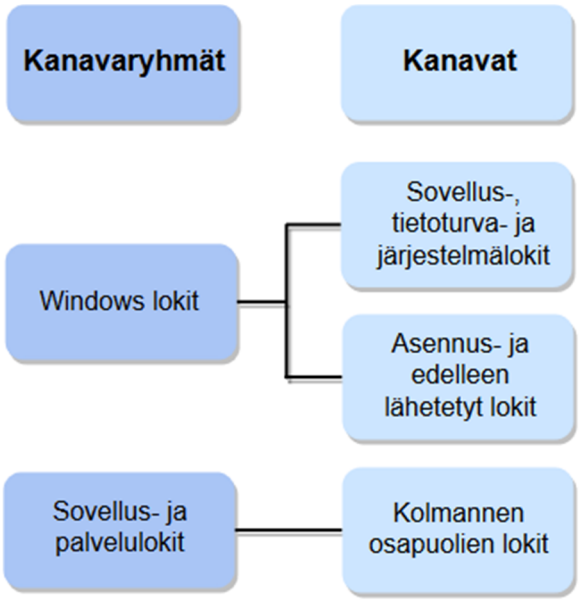
Kuva 6. SNMP-hallintaohjelman on lähetettävä kuittaus saapuneesta ilmoituksesta (Cisco n.d.).

SNMPv1:n lähettämiä trap-viestejä ei suojata, eikä niitä myöskään todenneta, joka altistaa ne samanlaisille heikkouksille kuin Syslogia käytettäessä. SNMPv3-protokollan lähettämissä viesteissä voidaan käyttää todennusta, joka tarjoaa suojaa väärennettyjä viestejä vastaan. (Chuvakin, Schmidt & Phillips 2012, 60.)

2.2.3 Windows Event Log

Windows Event Log (suom. tapahtumienvalvonta) on Windows-käyttöjärjestelmän sisäänrakennettua lokijärjestelmä. Tapahtumienvalvonta luo lokitietoa käyttöjärjestelmän sisäisestä toiminnasta, tietoturvasta, sekä ohjelmistojen tekemistä muutoksista. (Chuvakin, Schmidt & Phillips 2012, 62.)

Tapahtumienhallinta jakaa lokit useaan eri ryhmään. Näitä ryhmiä kutsutaan kanaviksi. Lokit tallennetaan kahden pääkanavaryhmän alle, jotka ovat Windows-lokit sekä sovellus- ja palvelulokit. Näiden kanavaryhmien alla on eri lokityypeistä muodostuvia kanavia. Kanavaryhmät ja niiden kanavat on havainnollistettu alla näkyvässä kuvassa 7. Windows-lokien alaisuuteen kuuluvat sovellus-, tietoturva-, järjestelmä-, sekä asennus- ja edelleen lähetetyt lokit. Oman kanavaryhmänsä muodostavat sovellus- ja palvelulokit, jonka alle kolmannen osapuolen ohjelmat voivat halutessaan viedä lokeja. (Charter 2008, 8.) Tapahtumienhallinta voi myös vastaanottaa lokitietoa muilta tietokoneilta. Tällöin tapahtumienhallinnan kautta on käynnistettävä tapahtumien keräyspalvelu ja luotava tilaus lokien vastaanottamiseksi. Tietokoneilta saapuvat lokit kerätään omaan tilauskanavaan. (Microsoft 2015.)

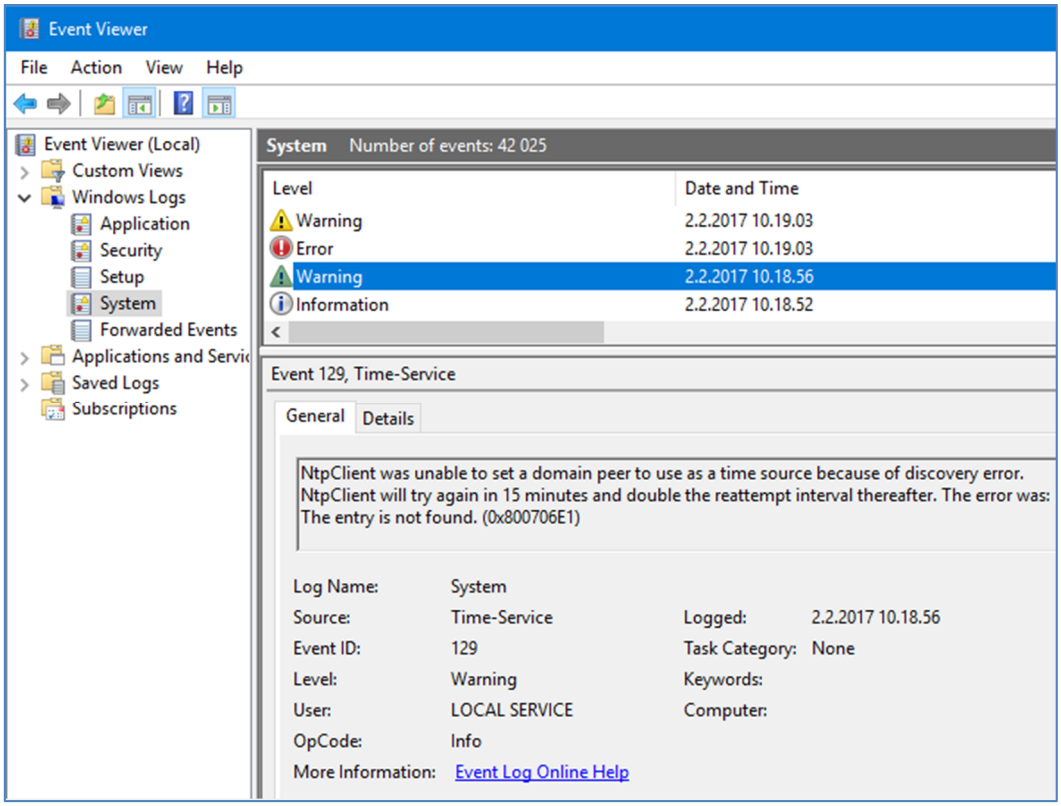


Kuva 7. Tapahtumienvalvonnan pääkanavaryhmät ja niiden kanavat (Charter 2008,8).

Tapahtumienvalvonta käyttää lokien merkitsemiseen useita tasoja, joilla se ilmaisee lokitapahtuman vakavuuden. Järjestelmä- ja sovellustason lokityypit ilmaistaan seuraavasti:

- tiedotus
- varoitus
- virhe
- kriittinen virhe.

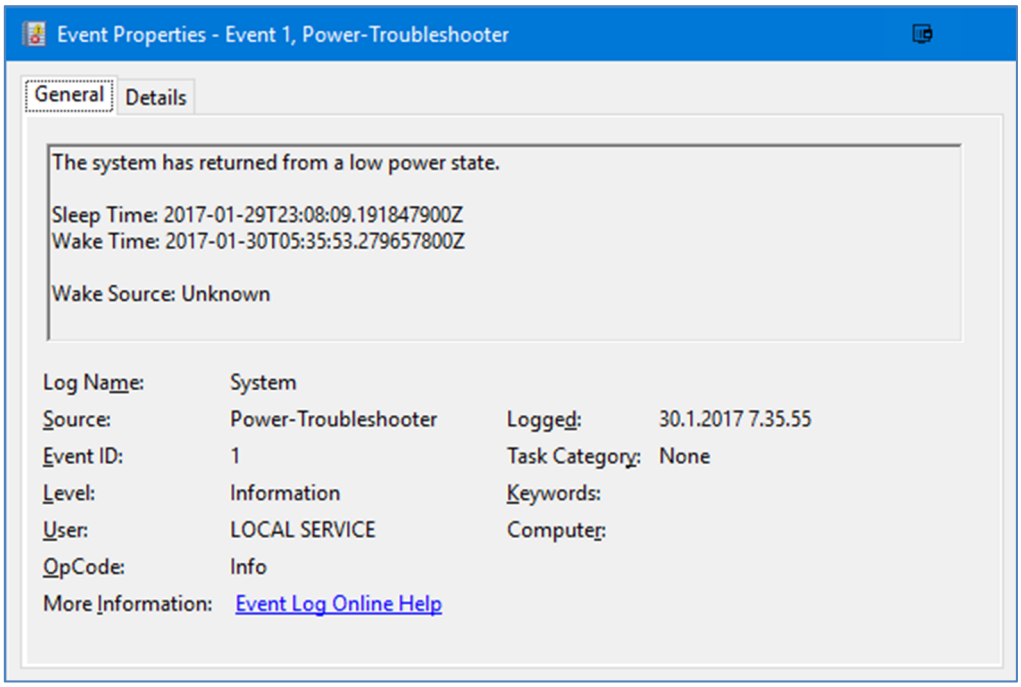
Tiedotusviesti kertoo sovelluksen tai järjestelmäosan muutoksesta, esimerkiksi palvelun käynnistämisestä. Varoitusviesti kertoo ongelmatilanteesta, jolla voi olla vaikutus palvelun toimintaan. Varoitusviestiin reagoimatta jättäminen voi johtaa suurempaan ongelmaan. Virheviesti kuvaa vikatilannetta, joka voi vaikuttaa ohjelman ulkoiseen toimintaan. Kriittinen virheviesti kertoo ohjelman kohtaamasta vikatilanteesta, josta se ei pystynyt palautumaan. Tietoturvalokissa on lisäksi vielä kaksi omaa tasoa, jotka viestivät käyttäjäluvan hyväksymisestä tai hylkäämisestä. (Microsoft n.d.) Kuvassa 8 on nähtävissä tapahtumanhallintapaneeli ja varoitus-tyypin lokiviesti.



Kuva 8. Tapahtumienhallintapaneeli ja varoitusloki (Kuvaruutukaappaus).

Tapahtumienvalvonnan lokit käyttävät XML-rakennetta viestin runkona. XML:n käyttäminen mahdollistaa siistin ja järjestyksellisen ulkoasun, mikä taas selkeyttää lokien lukemista. (Charter 2008, 8.)

Kuvassa 9 on esimerkki järjestelmän kirjaamasta lokista ja sen sisällöstä. Viestin sisältö viittaa järjestelmän palautumiseen lepotilasta. Lokiin on kirjattu aika, jolloin järjestelmä vaipui lepotilaan ja palasi sieltä. Viestistä näemme, että tieto on peräisin järjestelmän sisäisestä virranhallinnan palvelusta, joka suoritettiin paikallisella palvelutilillä. Lokista selviää myös viestin tyyppi, joka on tässä tapauksessa tiedotus, joka ei vaadi käyttäjältä toimia.



Kuva 9. Tapahtumienhallinnan keräämän lokiviestin sisältö (Kuvaruutukaappaus).

Jokaiselle tapahtumanvalvonnan kirjaamalle lokilla on oma Event ID-numerotunniste, jolla tietty tapahtuma voidaan tunnistaa. Event ID:n tarkoitus on helpommin tunnistaa vikatilanteet, joka nopeuttaa järjestelmänvalvojan tai IT-tuen vianselvitysprosessia. (Charter 2008, 5.)

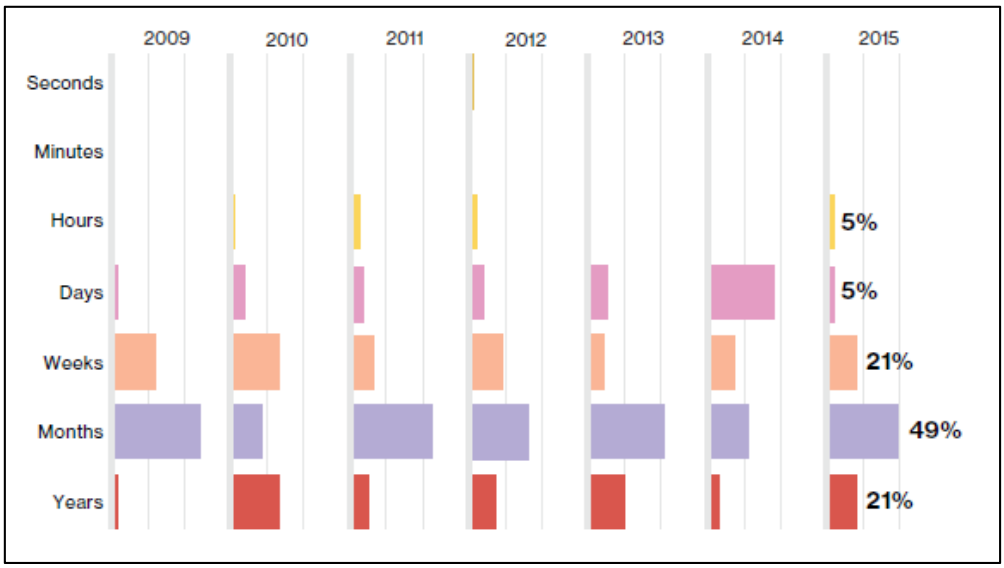
Varghese (2013) esitteli Microsoftin julkaisemassa blogissa ohjeistuksen, kuinka ryhmäkäytäntöjen ja paikallisten käytäntöjen avulla toimialueen työasemille voidaan ottaa käyttöön käyttäjien yksityiskohtaisempi lokitus, jonka avulla tapahtumienvalvonta kirjaa tarkasti työntekijöiden aktiviteetit heidän työasemillaan. Samalla Varghese pyytää blogissaan tarkistamaan lokivalvonnan laillisuus, ennen tarkan lokituksen käyttöönottoa, sillä tiedon keräämistä koskevat lait voivat vaihdella maasta tai toimialasta riippuen.

2.3 Lokien varastointi

Viestintävirasto (2016, 5) ohjeistaa säätämään lokien säilytysajan mieluummin riittävän pitkäksi kuin liian lyhyeksi. Lokien riittävä säilytysaika vaihtelee säilytettävän lokitiedon mukaan, joka voi olla kuudesta kuukaudesta 24 kuukauteen. Viestintävirasto neuvoo samalla arvioimaan tarvittavan lokitilan määrän kuukauden keskimääräisen lokikertymän mukaan.

Lokien säilytyksessä on otettava huomioon lainsäädäntö, joka voi asettaa vaatimuksia lokien säilytysajalle etenkin maksuliikenteestä puhuttaessa. Lokien säilytysaikaan vaikuttaa myös yrityksen oma linjaus, johon voi liittyä yrityksen ulkoiset sekä sisäiset riskitekijät. Etenkin sisäisissä tutkinnoissa pitkäaikaisesta lokien varastoinnista voi olla apua, sillä tietoturvamurron tapahtumasta voi pisimmillään kulua vuosia ennen kuin asia huomataan. Eri lokilähteiden kuten palomuurien, palvelinten ja verkkolaitteiden generoimat lokien ja lokityyppien säilytysiät on arvioitava tarpeiden ja riskitekijöiden mukaisesti, jolloin myös lokien varastointikustannuksissa voitaisiin järkevästi säästää. (Chuvakin, Schmidt & Phillips 2012, 72.)

Lokien lyhytikäisestä varastoinnista voi aiheutua ongelmia, jos yritykselle tapahtunutta tietomurtoa on selvitettävä pidemmältä ajalta, jolloin voidaan huomata, että lokitietoja tapahtumasta ei enää ole (Chuvakin, Schmidt & Phillips 2012, 237). Verizonin (2016, 38) teettämän vuosittaisen DBIR-raportin (Data Breach Investigations Report) mukaan vuonna 2016 tapahtuneista yritysten työntekijöiden oikeuksien väärinkäytöistä 49 % huomattiin kuukausien jälkeen, kun taas 21 % tapauksista huomattiin vasta useiden vuosien jälkeen (Kuva 10). Edellisvuosien raportit kertovat samanlaisesta jatkumosta. Vuonna 2016 tutkimukseen osallistui 67 yritystä.



Kuva 10. Tilasto osoittaa ajan, jonka sisällä yritysten työntekijöiden sisäiset väärinkäytöt ovat paljastuneet (Verizon 2016, 38).

Lokien varastointitapoja on monia jotka vaihtelevat hinnan, kapasiteetin ja käytettävyyden mukaan. Tuoreita ja tärkeitä lokeja voidaan tallettaa kiintolevyille tai pilvipalveluihin, jolloin niihin pääsee käsiksi nopeasti. Lokeja voidaan arkistoida myös esimerkiksi nauhalle tai optiselle medialle, jolloin tietoon käsiksi pääseminen voi olla hieman työläämpää, mutta kustannustehokkaampaa. (Chuvakin, Schmidt & Phillips 2012, 72.)

2.4 **Lokivalvonnan tasot**

Yrityksen tai organisaation tyypillinen toimintatapa lokienhallintaa kohtaan on reagoiva (Grimes 2012, 2). Reagoivat yritykset tarkastelevat lokitietoja tapahtuman jälkeisessä tilanteessa, jolloin tietoturvamurto tai vika on jo päässyt tapahtumaan (Chuvakin, Schmidt & Phillips 2012, 236). Tällä tasolla lokeja hyödynnetään vain reaktiona ongelmaan, jolloin järjestelmien vikatapahtumat yleensä etenevät käyttäjän tietoisuuteen. IT-tuki saa vikatapahtumasta viestin käyttäjän luoman tiketin avulla, jolloin vikatilanne on jo eskaloitunut kriittiseksi ja se vaikuttaa käyttäjän toimintaan häiritsevästi.

Reagoiva toimintaympäristö nostaa huolen myös yrityksen tietoturvan valvonnasta. Verizonin (2015, 11) mukaan yritykset, jotka osoittavat heikkoa lokien käyttöä ja valvontaa eivät todennäköisesti huomaa olevansa tietomurron kohteena, jolloin hyökkääjällä on enemmän aikaa tehdä tuhoja järjestelmässä. Useat tietoturvamurroista kärsineet yritykset eivät välttämättä havaitse murtoa lainkaan vaan saavat tietää siitä virkavallan tai esimerkiksi kolmannen osapuolen kautta, heidän suorittaessa yritykselle tietoturvatarkastusta (Verizon 2015, 59).

Ennakoivaa lokivalvontaa hyödyntävä yritys käyttää lokijärjestelmää varhaisena hälytysjärjestelmänä. Tällä tasolla yritys seuraa aktiivisesti eri järjestelmien toimintaa ja vertaa sitä aiempaan tapahtumahistoriaan. Järjestelmissä käytetään automatisoituja hälytysviestejä, jotka viestivät järjestelmänvalvojille mahdollisista poikkeavuuksista. Järjestelmän tilaa seuraamalla mahdollisiin ongelmiin voidaan reagoida välittömästi, ennen kuin ne kasvavat suuremmiksi. Esimerkiksi kiintolevyjen vikaantumisen ensimerkkejä voidaan seurata viallisia sektoreita laskemalla. Viallisten sektorien ilmetessä kiintolevy voidaan vaihtaa välittömästi uuteen, ennen kuin tietokone ensimmäisen kerran kaatuu tai osoittaa epävakaata käytöstä. (Grimes 2012, 2.)

2.5 **Lokitietojen kerääminen**

Käyttäjätietojen keräystä koskevat lakisäädännöt voivat vaihdella maiden ja eri toimialojen mukaisesti. Esimerkiksi Suomessa sosiaali- ja terveydenhuollon toimialalla on lain mukaan ylläpidettävä rekisteriä asiakastietojen käsittelystä. Tällöin henkilötietojen käsittelystä jää järjestelmään jälki, joka helpottaa valvontaa ja virantoimituksessa tapahtuneiden rikkeiden selvitystä. (Männikkö 2008.)

Valtiovarainministeriön (2009, 21–24) teettämän lokiohjeen mukaan organisaatiolla pitää olla perusteltu tarve lokien keräämiselle sekä ennalta määritelty suunnitelma tiedon keräyksestä. Organisaatiolla pitää myös olla ennalta sovitut toimintatavat ja järjestelmät lokien käsittelyä varten sekä suunnitelma lokien säilytykselle ja poistolle. Lokitiedot muuttuvat henkilörekisteriksi, jos lokien tieto sisältää henkilötietoja, josta henkilö voidaan tunnistaa. Henkilötietolaki (1999/523 §3.)

Henkilörekistereitä koskevat omat vaatimukset ja lainsäädännöt henkilötietojen suojelemiseksi, jotka organisaation pitää tiedostaa tämän tason tietoja kerätessä. Esimerkiksi lain mukaan rekisterin ylläpitäjän on huolehdittava, ettei asiattomilla ole pääsyä rekisteriin. Rekisteriä varten on myös oltava tekniset toimet, joilla sitä voidaan suojella tiedon muutoksilta, poistamisilta tai siirtämiseltä. (Henkilötietolaki 1999/523 §32.) Yksi näistä toimista voi olla tiedon varmuuskopiointi.

Valtiovarainministeriön (2009, 48) mukaan lokien käsittely on sallittua seuraaviin käyttötarkoituksiin:

- palvelujen toteuttamiseen
- laskutukseen
- markkinointiin
- tekniseen kehittämiseen
- maksullisen palvelun väärinkäyttötapauksissa
- tekniseen vianselvitykseen
- tietoturvallisuuden parantamiseen.

Kasvi (2014) kertoi Ylen haastattelussa, kuinka sosiaalisen median palvelut myyvät käyttäjien tietoja eteenpäin mainostajille ja markkinatutkimuksiin. Käyttäjätietoja voidaan kerätä esimerkiksi seurantaevästeiden avulla. Seurantaeväste on ohjelma, jonka avulla verkkosivulla vieraileva käyttäjä voidaan tunnistaa. Verkkosivulle mentäessä käyttäjän tietokoneelle ladataan seurantaeväste, joka pitää kirjaa millä sivuilla käyttäjä on käynyt ja mitä mainosbannereita käyttäjälle on näytetty. Tämän tiedon avulla käyttäjän verkkokäyttäytymistä ja mieltymyksiä voidaan seurata ja hänelle voidaan tarjota kohdennettuja mainoksia. (F-Secure, n.d.)

Käyttäjätiedon keräys ja sen hyödyntäminen käyttäjien profiloinnissa on kiistelty aihe. Käyttäjien profiloinnilla tarkoitetaan automatisoitua henkilötiedon keruuta, jonka tarkoitus on arvioida henkilön ominaisuuksia analysoimalla henkilön kiinnostuksen kohteita tai mieltymyksiä (Aalto-Setälä, 2016). Käyttäjätiedon profilointi koostuu kolmesta vaiheesta**.** Palvelun tarjoaja kuten Facebook ja Google kerää käyttäjistään tietoa, analysoi tiedon tai välittää sen eteenpäin yrityksille, jotka ovat erikoistuneet henkilödatan keruuseen ja louhintaan. Nämä yritykset jalostavat keräämästään tiedosta tarkkoja käyttäjäprofiileja ja käyttäytymismalleja. Tieto kaupataan eteenpäin mainostajille ja markkinointiin, jotka voivat tiedon avulla tarjota käyttäjälle kohdistettua mainontaa, josta hän voisi olla kiinnostunut. (Kosola, 2016.)

EU julkaisi vuonna 2016 uuden tietosuoja-asetuksen, jossa korostettiin yksilön oikeuksia henkilötietojen käsittelyssä. Uusi asetus tuo ihmisille jatkossa oikeuden kieltäytyä automatisoidusta profiloinnista. Asetus astuu voimaan vuonna 2018. (Warma & Luomala, 2016.)

3 KÄYTÄNNÖN OSUUDEN ESIVALMISTELU

3.1 Tietoa Elastic-ohjelmista

Elastic-ohjelmat perustuvat avoimeen lähdekoodiin. Tämä mahdollistaa sen, että niitä voi vapaasti käyttää ja muokata omiin käyttötarkoituksiin sopivaksi. Tässä osiossa tutustutaan Elastic-ohjelmistopinoon, joka koostuu kolmesta ohjelmasta: Logstash, Elasticsearch ja Kibana. Lisäksi käsitellään kahta Beat-ohjelmaa, joiden tehtävä on lähettää lokitietoa asiakaskoneilta Elastic-palvelimelle.

3.1.1 Elasticsearch-hakumoottori

Elasticsearch on avoimeen lähdekoodiin perustuva haku- ja analytiikkamoottori. Elasticsearch on erittäin skaalautuva ja mahdollistaa tiedon talletuksen, haun ja suuren tietomäärän analysoinnin reaaliajassa. (Chhajed 2015.) Elasticsearchia voidaan ajaa kannettavalta tai se voidaan skaalata satoihin palvelimiin ja sillä voidaan käsitellä jopa tuhansia terabittejä tietoa (Gormley & Tong 2015, 1)**.** Elasticsearch voidaan integroida muihin sovelluksiin ja tästä syystä sitä käytetäänkin yleisesti sovellusten alla toimivana hakumoottorina (Chhajed 2015).

Elasticsearch on hyvin joustava, ja suuret verkkopohjaiset yritykset käyttävät sitä moniin eri tehtäviin. Pienemmässä mittakaavassa sitä voidaan käyttää verkkokaupan hakukoneena, jolla käyttäjät voivat etsiä verkkokaupasta tuotteita. Sitä käyttävät myös suuryritykset kuten Wikipedia, joka käyttää Elasticsearchia Wikipedian hakukonemoottorina. Elasticsearchin avulla Wikipedia tarjoaa käyttäjilleen tekstihakua sekä artikkeliehdotuksia. (Gormley & Tong 2015, 1) Muihin ohjelman käyttäjiin kuuluvat muun muassa GitHub, SoundCloud ja Netflix (Chhajed 2015). Opinnäytetyössä Elasticsearchia käytetään lokihallintajärjestelmän hakumoottorina, jonka tarkoitus on indeksoida ja varastoida tietoa, jotta sitä voidaan hakea luettavaksi verkkopohjaisesta käyttöliittymästä.

Elasticseach on kehitetty Apachen Lucene-hakukonemoottorin pohjalta. Lucenen avulla tietoa voidaan dynaamisesti indeksoida etukäteen ennen kuin sen rakenne on selvillä. Indeksoinnin avulla Elasticsearch kykenee nopeasti etsimään haettua tietoa muun tiedon joukosta. (Chhajed 2015.) Turnbull (2016, 33) kuvaa indeksiä kirjan hakemistoksi, jonka avulla kirjasta voidaan etsiä tiettyä sanaa. Haetun sanan kohdalta voidaan nähdä sivunumero, johon kirjassa viitataan. Tämä tarkoittaa sitä, että Elasticsearch ei suoraan hae tekstiä muun tekstin joukosta vaan ensin indeksoi tekstin, jolloin se voi hakea tiettyä sanaa indeksistä, joka paljastaa missä haettu tieto sijaitsee. Tästä johtuen indeksointi nopeuttaa tiedon etsintää.

Elasticsearch käyttää ohjelmointirajapintana RESTful API:a (Application Programming Interface), jolle ohjelman kyselyt voidaan kohdistaa. Ohjelmaa voidaan käyttää esimerkiksi Javascript-, Perl-, PHP- ja Ruby-ohjelmointikielenrajapinnoilla. (Gormley & Tong 2015, 7.) Elasticsearchin muihin ominaisuuksiin kuuluvat monikielinen haku, geopaikannus, automaattinen täydennys sekä kontekstipohjaiset ehdotukset. Tämän lisäksi ohjelmaan voidaan luoda omia lisäosia. Palvelun voi integroida myös pilvipohjaiseen infrastruktuuriin, kuten Amazon Web Servicesiin (AWS). (Chhajed 2015.)

3.1.2 Logstash-lokienkäsittelyohjelma ja Kibana-käyttöliittymä

Logstash on lokienkäsittelyohjelma, joka vastaanottaa tietoa useista eri lähteistä ja muuttaa sen samaan formaattiin. Logstashin rakenne on kolmiosainen. Ensin Logstash vastaanottaa syötteen lähteestä, joko suoraan tai Beat-ohjelman avulla, joka lähettää lokitiedon Logstashille. Tämän jälkeen Logstash suodattaa lokitiedon ja jäsentää siitä kerätyn tiedon samaan formaattiin sopivaksi. Näin lokeista muodostuu samanmuotoisia ja helposti haettavia, jolloin myös lokien analysoinnista tulee helpompaa. Lopuksi Logstash lähettää muutetun lokitiedon eteenpäin toisen ohjelman käsiteltäväksi, joka voi olla muun muassa erillinen monitorointiohjelma tai indeksointiohjelma. (Turnbull 2016, 8–9.)

Logstash voi vastaanottaa syötteitä useista eri lähteistä TCP- ja UDP-protokollien välityksellä. Verkkolaitteilta voidaan kerätä tietoa SNMP-protokollan avulla. Logstash voi vastaanottaa ja prosessoida Windows Event Login tietoa sekä Linux- ja Unix-järjestelmien syslog-lokitietoa. Logstash kykenee myös käsittelemään eri tietokannoista saapuvaa tietoa. Logstash on ohjelmoitu JRuby-ohjelmointikielellä ja se toimii Java Virtual Machine (JVM) -alustalla. (Turnbull 2016, 8–9.)

Kibana on selainpohjainen tiedon visualisointiin ja analysointiin tarkoitettu alusta. Kibana on luotu HTML:n ja Javascriptin avulla ja se on suunniteltu toimimaan yhdessä Elasticsearchin kanssa. Elasticsearchin prosessoimaa tietoa voidaan seurata reaaliajassa Kibanan hallintapaneelista. Kibanassa kerätyn tiedon avulla voidaan luoda graafisia taulukoita, kaavioita ja karttoja, jotka helpottavat hallittavan ympäristön kokonaiskuvan hahmottamista. Kibana on yhteydessä Elasticsearchiin RESTful API-ohjelmistorajapinnan kautta. Kibanassa voidaan luoda hallintapaneeleja, jotka ovat muokattavissa drag-and-drop-ominaisuuksien avulla. Hallintapaneelit voidaan integroida eri järjestelmien käytettäväksi. (Chhajed 2015.)

3.1.3 Filebeat- ja Winlogbeat-lokienlähetysohjelmat

Beatit ovat Elasticin kehittämiä kevyitä lisäohjelmia, joiden tarkoitus on lähettää asiakaspalvelimen tai työaseman lokitieto Logstashin prosessoitavaksi. Beat-ohjelma asennetaan kohdejärjestelmään, jonka lokitietoa halutaan lähettää. Beat-ohjelmistoperheeseen kuuluu useita eri ohjelmia, jotka soveltuvat erilaisen tiedon lähettämiseen. Opinnäytetyön käytännön osuudessa käytetään kahta eri Beat-ohjelmaa, jotka ovat Filebeat ja Winglogbeat. Filebeat tullaan asentamaan Linux-pohjaisiin käyttöjärjestelmiin, kun taas Winlogbeat asennetaan vastaavasti Windows-pohjaisiin käyttöjärjestelmiin. (Elasticsearch n.d.)

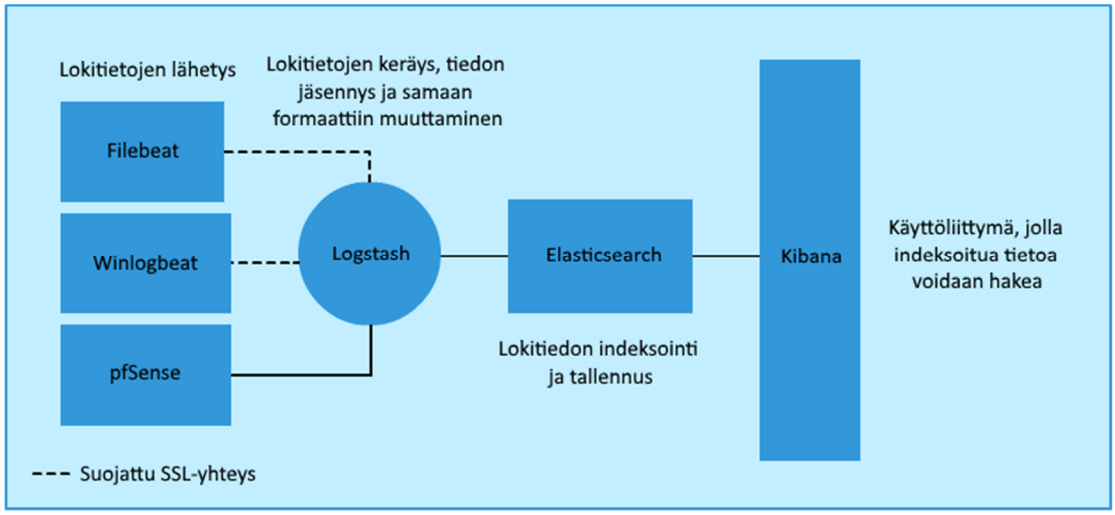
Filebeatilla on kaksi pääosaa jotka ovat etsijät (engl. prospectors) ja kerääjät (engl. harvesters). Kerääjän tehtävä on lukea tiedoston sisältöä ja lähettää se eteenpäin ulostulolle. Jokaista tiedoston prosessointia varten luodaan oma keräilijä. Keräilijä pystyy yhä lukemaan tiedostoa, jos se poistetaan ennen kuin keräilijä on sulkenut tiedoston. Etsijä huolehtii keräilijöistä ja etsii samalla luettavia lokitiedostoja. Filebeatin konfiguraatiotiedostossa määritetään, mitä lokeja ja lokityyppejä järjestelmästä haetaan luettaviksi. Etsijä etsii tämän perusteella luettavat tiedostot ja käynnistää kerääjän läpikäymään lokitiedostoa. (Elasticsearch n.d.)

Filebeat seuraa tiedostoja ja päivittää niiden tilan rekisteritiedostoon. Tilan seurannan avulla tiedetään, milloin viimeksi keräilijä on lukenut tiedostoa, jolloin voidaan varmistaa, että kaikki tiedoston lokitiedot on lähetetty. Filebeat pitää kirjaa siitä, mitkä ovat olleet viimeiset lähetetyt lokirivit, jolloin lokilähetyksen katketessa Filebeat tietää mistä jatkaa. (Elasticsearch n.d.)

Winlogbeat- ja Filebeat-ohjelmat eroavat toisistaan niiden käsittelemien lokityyppien takia. Winlogbeat voi lähettää Windowsin tapahtumienhallinnan lokitapahtumia Elasticsearchille tai Logstashille. Winlogbeat lukee tapahtumienhallinnan lokitapahtumia Windows API-ohjelmistorajapinnan avulla. Filebeatin tavoin Winlogbeat suodattaa tapahtumia käyttäjän määrittämien asetusten mukaisesti, jonka jälkeen tieto lähetetään lokienkerääjälle. Winlogbeat seuraa lokien lukuprosessejaan samalla tavalla kuin Filebeat, joten ohjelman uudelleenkäynnistyksen jälkeen se tietää mitä lokitietoa on viimeksi käsitelty ja jatkaa siitä. (Elasticsearch n.d.)

3.2 Suunnitelma

Lähiverkkoon asennetaan Logstashista, Elasticsearchista ja Kibanasta koostuva ohjelmistopino. Elastic-pinon jokaisella ohjelmalla on oma roolinsa. Yhdessä niistä voidaan rakentaa lokihallintajärjestelmä, joka vastaanottaa lähiverkon tietokoneiden ja laitteiden lokit ja muuttaa ne samaan formaattiin, tallentaa ja indeksoi ne (Kuva 11). Asiakaspalvelimiin asennetaan Beat-ohjelma, jolla lokitiedot lähetetään Elastic-palvelimelle. Beat-ohjelman ja Elastic-palvelimen välinen yhteys suojataan SSL-sertifikaatin avulla. pfSensen lokiyhteyden suojaamiseen ei tässä opinnäytetyössä käsitellä. Lokien prosessointiin käytetään Logstashia, joka kerää, suodattaa ja muuttaa eri lähteistä vastaanotetun lokitiedon samaan formaattiin. Elasticsearchin tehtävä on indeksoida ja tallentaa lokitiedot, jolloin niitä voidaan hakea Kibana-verkkokäyttöliittymän avulla. Lopuksi testataan Kibanan visualisointityökaluja luomalla pfSensen ja palvelimien lokitiedosta graafisia esityksiä.



Kuva 11. Elastic-ohjelmien roolitus lokihallintajärjestelmässä.

Elastic-ohjelmat valittiin, koska ne ovat avoimeen lähdekoodiin perustuvia ja niiden käyttö on ilmaista, joka on suotavaa asennusympäristö huomioon ottaen. Elasticin verkkosivuilta löytyi hyvät ja ajan tasalla olevat ohjeet Elastic-ohjelmien asennukseen. Käyttäjien julkaisemien asennusohjeiden kanssa pitää olla tarkkana, sillä ne voivat sisältää vanhentunutta tietoa ohjelmien aiempiin versioihin liittyen. Elastic-ohjelmien suosiosta on hyötyä myös ongelmatilanteissa, jolloin apua on helpompi löytää.

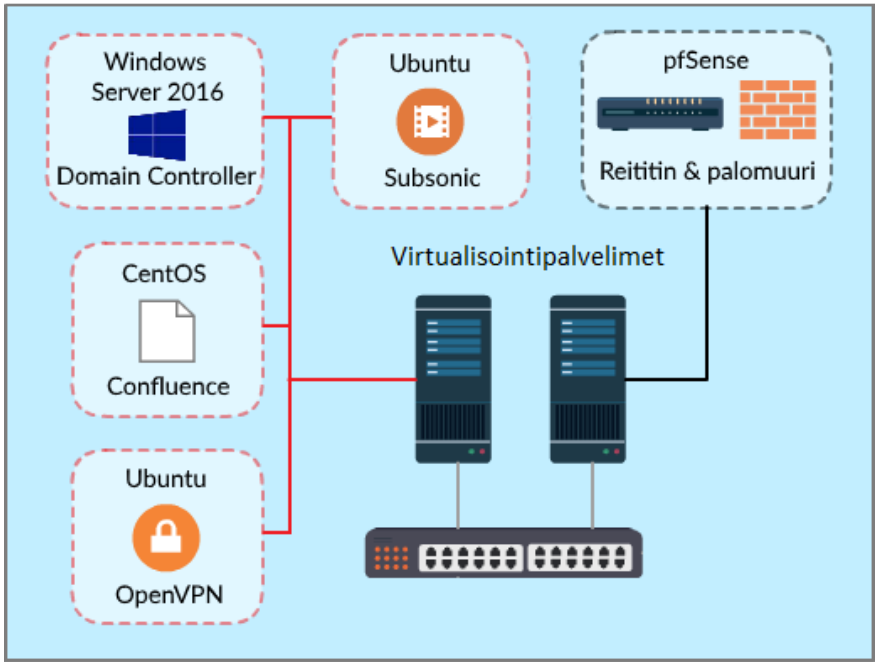
Käytännön osuuteen valmistauduttiin tutustumalla ohjelmistokehittäjien asennusoppaaseen. Elasticin (n.d.) oppaassa on selkeästi selostettu ohjelmien asennusprosessi. Oppaassa suositellaan asentamaan Elastic-pinon ohjelmat tietyssä järjestyksessä aloittaen ensin Elasticsearchista. Tämän jälkeen asennetaan Kibana ja lopuksi Logstash sekä Beat-ohjelmat. Asennusjärjestys takaa sen, että vaadittavat osat ovat toiminnassa ennen kuin ne yrittävät hakea tai luoda yhteyksiä toisiinsa.

3.3 Toteutusympäristö

Opinnäytetyön käytännön osuuden toteutusympäristönä toimii kotiverkko. Kotiverkossa on käytössä useita virtuaalipalvelimia sekä pfsense-reititin, joiden valvontaa halutaan parantaa. Toteutusympäristössä ei ole ollut aiempaa alustaa eri järjestelmien valvonnalle. Virtuaalipalvelinten ja reitittimen monitorointi on ollut tähän asti haastavaa ilman yhtenäistä valvontajärjestelmää ja tähän haluttaisiin ratkaisu.

Etenkin suuren kokonaiskuvan hahmottaminen on ollut toimintaympäristössä haastavaa ja se tulee vaikeutumaan ympäristön kasvaessa. Lokien hyödyllisyys on käyttäjäympäristössä tiedostettu ja niitä on aiemmin käytetty järjestelmissä piilevien vikojen korjaamiseen. Siksi haluttaisiin, että järjestelmiä ja ohjelmistoja koskevaa yksityiskohtaista tietoa pystyttäisiin hyödyntämään paremmin. Lokihallintajärjestelmä parantaisi hallittavan ympäristön kokonaiskuvan hahmottamista sekä lisäisi käyttäjän tietoisuutta eri järjestelmien toiminnoista.

VMware-virtualisointipalvelimella on luotu virtuaalikone, johon Elastic-ohjelmistopino tullaan asentamaan. Lähiverkon virtuaalikoneisiin kuuluu Windows-toimialuepalvelimen lisäksi kolme Ubuntu-palvelinta sekä CentOS-palvelin (Kuva 12). Palvelimia käytetään eri palveluiden kuten OpenVPN:n, Subsonicin ja Confluencen ylläpitämiseen. Toimialueen toiselle virtualisointipalvelimelle on asennettu pfSense-reititin, joka toimii lähiverkon reitittimenä ja palomuurina. pfSensen palomuuri generoi tietoturvalokeja, jotka halutaan keskittää lokihallintajärjestelmän prosessoitavaksi. Käytännön työn loppuvaiheessa palomuurilokeista luodaan Kibanan avulla graafisia esityksiä, joiden avulla on tarkoitus visualisoida palomuurilta kerättyä tietoa.



Kuva 12. Verkkoympäristö johon projekti toteutetaan.

Elastic-palvelimen laitteistoresursseja määritettäessä otettiin huomioon Elasticin omat tuotantosuositukset, joissa neuvottiin käytettäväksi vähintään 8 GB keskusmuistia. Elastic-palvelimen laitteistoresursseiksi määritettiin 8 GB keskusmuistia, 2 CPU -ydintä sekä 35 GB SSD-kiintolevytilaa. Lokien viemää tilaa on tässä vaiheessa vaikea arvioida. Tätä onkin syytä seurata lokien keskittämisen jälkeen. Hyvä vaihtoehto tilan vähenemiseen olisi luoda lokeja varten erillinen 10 GB:n kiintolevyosio, jonka kapasiteetti koostuisi perinteisestä kiintolevytilasta, jolloin olemassa olevat lokit voitaisiin siirtää uuteen osioon. Elastic-palvelinta hallitaan SSH-etäyhteyden avulla.

Elastic-palvelimelle on asennettu palvelinversio Ubuntu 16.10-käyttöjärjestelmästä. Elastic-palvelimen käyttöjärjestelmä valittiin RedHat-pohjaisen CentOS:n ja Debian-pohjaisen Ubuntun väliltä. Ubuntuun päädyttiin, koska siitä oli enemmän käyttökokemusta. Palvelimella on UFW-palomuuri, joka konfiguroitiin valmiiksi. Palomuurin säännöt estävät kaiken saapuvan liikenteen ja sallivat lähtevän liikenteen. Elastic-palvelimelta avattiin kaksi porttia, jotta palvelin voi vastedes vastaanottaa lokeja lähiverkon palvelimilta sekä sallia yhteyksiä Kibanan verkkokäyttöliittymälle. Lokien vastaanottamiseen käytetään TCP-porttia 5044. Verkkokäyttöliittymän porttinumeroksi määritettiin 5601. Molemmat säännöt luotiin niin, että portteihin sallitaan yhteydet vain lähiverkon sisältä. Sallittaviin portteihin lisättiin myös 5140, joka vastaanottaa lokitietoa pfSense-reitittimen IP-osoitteesta. (Kuva 13.)

```
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
192.168.1.16 5044/tcp      ALLOW IN    192.168.1.0/24
192.168.1.16 5601/tcp      ALLOW IN    192.168.1.0/24
192.168.1.16 5140/tcp      ALLOW IN    192.168.1.1
192.168.1.16 5140/udp      ALLOW IN    192.168.1.1
192.168.1.16 22/tcp        ALLOW IN    192.168.1.0/24
```

Kuva 13. UFW-palomuurin porttiasetukset (Kuvaruutukaappaus).

4 ELASTIC-LOKIHALLINTAJÄRJESTELMÄN KÄYTTÖÖNOTTO

4.1 Elastic-ohjelmien asennus

Elasticsearch ja Logstash vaativat toimiakseen ympäristön johon on asennettu Java. Elasticin (n.d.) asennusohjeessa suositellaankin palvelimelle asennettavaksi uusin versio Javasta, joka on kirjoitushetkellä versio 8. Javaa asennettaessa ensin lisätään PPA-pakettivarasto apt-paketinhallinta-ohjelman latauslähteisiin. PPA-pakettivarasto on Canonical-nimisen yrityksen ylläpitämä ja Ubuntun pakettivarastosta poikkeava pakettivarasto, jolla käyttäjät ja yritykset voivat levittää ohjelmistoja ilman, että ne kulkevat Ubuntun virallisen pakettivaraston kautta (Hoffman 2015). Tämän jälkeen apt-paketinhallinta päivitettiin ja sen avulla asennettiin Java.

Elasticin (n.d.) asennusohjeessa suositellaan asentamaan Elastic-pinon ohjelmat tietyssä järjestyksessä, jotta ohjelmat löytäisivät toisensa ja voivat luoda heti yhteyksiä. Ensimmäisenä asennettiin hakukonemoottori Elasticsearch. Elastic on suojannut kaikki asennuspaketit PGP-avaimella. Purkamista varten Elasticin sivuilta ladattiin julkinen PGP-avain, joka lisättiin suoraan apt-paketinhallintaan. Tämän jälkeen luotiin Elastic-pakettivarasto paketinhallintaan. Uuden pakettivaraston avulla apt-paketinhallinta tietää verkkosivuosoitteen, josta Elastic-ohjelmat löytyvät.
Lopuksi paketinhallinta päivitettiin ja sen avulla asennettiin Elasticsearch.

Elasticsearch lisättiin automaattisesti käynnistyväksi prosessiksi systemd:n avulla. Systemd huolehtii siitä, että tarvittavat prosessit, ajurit ja ohjelmat käynnistyvät tietokoneen käynnistyksen yhteydessä, ja sitä voidaan käyttää ohjelmien hallinnoimiseen. Seuraavaksi asennettiin verkkohallintapaneeli Kibana. Koska aiemmassa vaiheessa paketinhallintaan lisättiin Elastic-pakettivarasto, Kibana voitiin nyt hakea ja asentaa suoraan apt-paketinhallinnan kautta. Asennuksen jälkeen Kibana käynnistettiin ja se lisättiin automaattisesti käynnistyväksi ohjelmaksi. Viimeisenä Elastic-pinon ohjelmana asennettiin lokien vastaanottamiseen ja prosessointiin tarkoitettu Logstash. Kaikista kolmesta ohjelmasta asennettiin 5.2-versiot.

Elastic on muuttanut ohjelmiensa versionumerointia versiosta 5.0 lähtien. Nykyisin kaikki Elastic-ohjelmat päivitetään samaan aikaan, jolloin ohjelmien versionumeroinnit pysyvät yhtenäisinä. Uusi versionumerointi on luotu selkeyttämään eri Elastic-ohjelmien välistä yhteensopivuutta. Eri Elastic-ohjelmia asennettaessa kannattaakin huomioida, että ohjelmien versionumeroinnit täsmäävät. (Elasticsearch n.d.)

4.2 **Elastic-palvelimen konfigurointi**

Palvelimelle asennettiin edellisessä luvussa Elastic-ohjelmat. Seuraavaksi asennetut ohjelmat konfiguroidaan. Konfigurointiin kuuluu yhteysasetuksien määrittäminen. Yhteysasetuksien määrittäminen on tärkeää, jotta Elastic-ohjelmien tiedonkulku saadaan toimimaan ohjelmasta toiseen. Yhteysasetuksiin kuuluu myös asiakaskoneen ja Elastic-palvelimen välisen yhteyden suojaus. Tätä varten luodaan SSL-sertifikaatti ja yksityinen avain.

Yhteysasetuksien lisäksi kaikissa Elastic-ohjelmissa on niiden toimintaan liittyviä asetuksia joita pitää säätää, jotta ohjelma saadaan toimimaan halutulla tavalla. Logstashin konfigurointi on erityisen tärkeää koska, sillä määritetään, miten vastaanotettuja lokeja prosessoidaan ennen niiden lähettämistä indeksoitavaksi.

4.2.1 Elasticsearchin indeksointipohja ja asetukset

Elasticsearch käyttää tiedon analysointiin indeksointipohjaa. Indeksointipohjan avulla Elasticsearch tietää miten sen kuuluu käsitellä eri tietokenttiä. (Elasticsearch n.d.) Indeksointipohja ladattiin Elasticsearchiin curl-työkalun avulla. Curl on avoimeen lähdekoodiin perustuva työkalu, jolla tietoa voidaan siirtää tai hakea palvelimelta (Stenberg n.d.).

Elasticsearch on suunniteltu toimimaan oletusasetuksilla, mutta on hyvä kuitenkin huomioida muutama tärkeä asetus, joita voi tarvita isommassa tuotantoympäristössä. Elasticsearchin muutokset tehdään */etc/elasticsearch*-polusta löytyvään *elasticsearch.yml*-nimiseen konfiguraatiotiedostoon.

Elasticsearch luo asennusvaiheessa palvelimesta klusterin ja samalla verkon pisteen (node). Elasticsearch-klusteri koostuu yhteen tuoduista node-palvelimista, jotka varastoivat tietoa ja osallistuvat yhdessä klusterin indeksointi- ja hakutoimintoihin. (Elasticsearch n.d.) Elasticsearch-klusteri koostuu oletuksena yhdestä node-palvelimesta, jos klusteriin halutaan liittää lisää palvelimia, on palvelinten *elasticsearch.yml*-konfigurointitiedoston *cluster.name*-asetukseen määritettävä klusterin nimi, johon palvelimet halutaan liittää. Konfiguraatiotiedostoon voidaan myös määrittää lista node-palvelimien IP-osoitteista, jotka halutaan liittää klusteriin. Tämä onnistuu listaamalla palvelimien IP-osoitteet *discovery.zen.ping.unicast.hosts*-komennon jälkeen. *Node.name*-asetuksesta voidaan vaihtaa node-palvelimen nimeä. Oletuksena palvelimille annetaan seitsemästä kirjaimesta muodostuva satunnaisesti generoitu ID-tunnus. (Elasticsearch n.d.)

Asetuksilla *path.data* ja *path.logs* määritetään, missä Elasticsearchin tietoaineisto ja lokit sijaitsevat. Elasticsearchin (n.d.) asennusoppaassa ke-

hotetaan muuttamaan näiden tiedostojen tallennuspolut, sillä niihin tallennetut tiedostot voivat joutua ylikirjoitetuksi ohjelman päivityksen yhteydessä. RPM- ja Debian-jakeluversiot käyttävät oletuksena muutettuja polkuja.

4.2.2 SSL-sertifikaatin ja avaimen luonti

Aiemmassa luvussa (2.2) tutustuttiin lokilähteisiin. Lokilähteitä tutkittaessa huomattiin niiden muodostavan tietyissä olosuhteissa mahdollisen tietoturvariskin. Suojaamattoman tietoliikenteen ongelmakohdat otettiin huomioon projektin suunnitteluvaiheessa. Tästä syystä Elastic-palvelimen ja asiakaskoneen välinen yhteys suojataan SSL-sertifikaatin avulla. SSL-yhteyden kautta kulkeva tieto suojataan suojausavaimen avulla. Yhteyden suojauksen ansiosta lokiyhteyden kaappaamisesta ja tiedon salakuuntelusta tulee vaikeampaa. Ongelmaksi muodostuvat pfSense-reitittimen lähettämät lokitiedot. Reitittimen verkkohallintapaneelissa ei ole vaihtoehtoa yhteyden suojaukselle. SSL-yhteyden muodostaminen on todennäköisesti mahdollista reitittimen komentorivin avulla, mutta tässä opinnäytetyössä ei siihen syvennytä.

SSL (Secure Sockets Layer) on sovellustason protokolla, jonka tarkoitus on suojata tietokoneiden välinen tiedonsiirto. SSL-sertifikaatilla palvelin todentaa itsensä asiakaskoneelle. Todentamisen jälkeen tietokoneiden välinen tietoliikenne suojataan yksityisavaimella, jolloin tietoa ei voi salakuunnella. (Ubuntu n.d.) SSL-yhteyden avulla herkkäluontoista tietoa voidaan siirtää turvallisesti asiakaskoneelta palvelimelle. SSL-suojausta käytetään yleisimmillään verkkokauppojen sekä verkkopankkien sivuilla.

SSL-yhteyden muodostaminen perustuu monivaiheiseen SSL- tai TLS-kädenpuristukseen. Asiakaskone luo ensin yhteyden Elastic-palvelimeen. Palvelimet vaihtavat keskenään tiedot käyttämistään SSL- tai TLS-versioista. Näin varmistetaan, että molemmat koneiden käyttävät protokollat ovat yhteensopivia. Näiden tietojen pohjalta luodaan ensiyhteys. Elastic-palvelin lähettää oman SSL-sertifikaatin sekä julkisen avaimen asiakaskoneelle, joka tässä tapauksessa tarkistaa sertifikaatin vertaamalla sitä omaan sertifikaattiin. Tämän jälkeen asiakaskone lähettää julkisella avaimella suojatun bittitunnisteen. Palvelin purkaa bittitunnisteen oman yksityisen avaimen avulla ja laskee siitä suojausavaimen, jolla koneiden välinen tietoliikenne suojataan. Näin tietokoneet voivat suojata toisilleen lähettämänsä tiedon ja purkaa sen omissa päissään. (IBM 2017.)

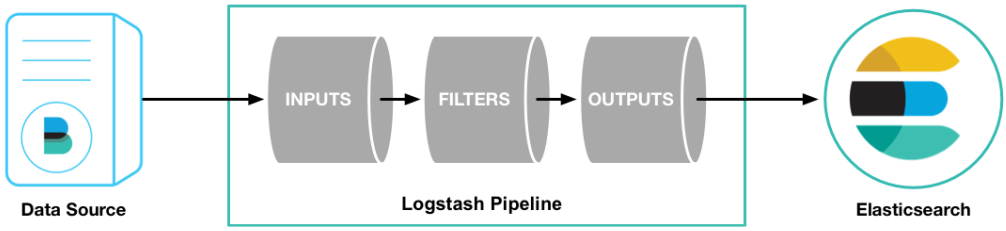
SSL-sertifikaatin ja suojausavaimen luomiseen käytettiin apuna OpenSSL-työkalua. OpenSSL on avoimeen lähdekoodiin perustuva komentorivityökalu, jolla voidaan luoda erityyppisiä SSL-sertifikaatteja ja avaimia. (Ubuntu n.d.) Sertifikaattia varten Elastic-palvelimen juurihakemistoon luotiin oma sertifikaattikansio. OpenSSL-työkalun yhteydessä käytettiin *openssl.cnf*-tiedostoa, joka sisältää valmiita oletusasetuksia sertifikaattien ja avaimien luomiseen. Tiedosto on hyödyllinen tilanteissa, joissa pitää

luoda eri sertifikaatteja ja avaimia useille eri tietokoneille. *Openssl.cnf*-tiedoston asetukset voidaan tuoda OpenSSL-työkalulle *-config* parametrin avulla. *Openssl.cnf*-tiedosto löytyy polusta */etc/ssl*. Tiedostoon muutettiin valmiiksi oikeat maakohtaiset asetukset sekä määritettiin *subjectAltName*-arvo, jolla määritettiin Elastic-palvelimen IP-osoite tunnistusta varten. Lopuksi SSL-sertifikaatti ja yksityinen avain luotiin OpenSSL-työkalulla komennolla: *sudo openssl req -config /etc/ssl/openssl.cnf -x509 -days 365 -batch -nodes -newkey rsa:2048 -keyout logstash.key -out logstash.crt*.

OpenSSL-komennolla haetaan ensin asetustiedosto, jonka pohjalta sertifikaatti luodaan. *X509*-parametri määrittää, mitä sertifikaattityyppiä käytetään. *Days*-parametrillä määritetään sertifikaatin voimassaoloaika. *Nodes*-parametrillä kerrotaan, ettei avainta tarvitse suojata erillisellä salasanalla. Samalla luodaan Elastic-palvelimelle 2048-bittinen RSA-suojattu avain. Sertifikaatista tehtiin kopio, joka siirrettiin valmiiksi asiakaspalvelimille. SSL-asetukset otetaan käyttöön seuraavissa vaiheissa asetuksia määritettäessä.

4.2.3 Logstashin konfigurointi

Asiakaskoneiden lokit lähetetään Logstashin käsiteltäväksi. Lokit kulkevat kolmiosaisen putkiston läpi, johon kuuluu tulo, lähtö ja suodattimet (Kuva 14). Tulo vastaanottaa tiedon lähettäjältä. Suodattimet suodattavat ja jäsentävät tietoa. Lopulta tieto lähetetään lähdön kautta Elasticsearchille tai muulle vastaavalle ohjelmalle indeksoitavaksi. Suodattimia ei ole pakko käyttää, mutta ne auttavat tiedon jäsentämisessä. (Elasticsearch n.d.)



Kuva 14. Tiedon kulku Logstashin läpi (Elasticsearch n.d.).

Logstash-putkiston tuloja, lähtöjä ja suodattimia voidaan muokata konfiguraatiotiedostojen avulla. Putkiston asetukset voidaan kirjoittaa samaan tiedostoon tai niitä varten voidaan luoda omat tiedostot. Konfiguraatiotiedostojen tallennuspolku on */etc/logstash/conf.d*. Tuloa, suodattimia ja lähtöä varten luotiin omat konfiguraatiotiedostot. Konfiguraatiotiedoston nimen eteen voi laittaa numeron, jolloin Logstash käsittelee ne numerojärjestyksessä.

Logstashin pfSense-osuuteen katsottiin mallia Elijahin (2015) julkaisemasta ohjeesta. Ensin luotiin tulolle konfiguraatiotiedosto nimellä *01-inputs.conf* (Kuva 15), jolla määritettiin tiedot lokeja lähettävien laitteiden

ja Elastic-palvelimen välisen yhteyden muodostamiseksi. Tiedostossa määritettiin Beat-ohjelmien sekä pfSense-reitittimen lähettämän lokitiedon vastaanottamista koskevat asetukset. Jokainen tulo eritellään muista. Beat-ohjelmien lähettämiä lokiviestejä varten varattiin porttinumero 5044. Beat-osiossa määritettiin myös SSL-yhteys ja ohjattiin polut aiemmassa luvussa (4.2.2) luodulle SSL-sertifikaatille sekä avaimelle. Reitittimen tieto otetaan vastaan TCP:n ja UDP:n kautta. pfSense-reititin käyttää syslogia lokitietojen generoimiseen. pfSensen lähettämä tieto vastaanotetaan portista 5140.

```
input {
    beats {
        port => "5044"
        ssl => true
        ssl_certificate => "/certificate/logstash.crt"
        ssl_key => "/certificate/logstash.key"
    }
}
input {
    tcp {
        type => "syslog"
        port => 5140
    }
}
input {
    udp {
        type => "syslog"
        port => 5140
    }
}
```

Kuva 15. Logstashin tulo-asetukset (Kuvaruutukaappaus).

Suodattimien avulla lokitietoa voidaan suodattaa ja jäsentää. Tätä varten voidaan käyttää grok-suodatintyökalua, joka tulee Logstashin mukana. Grok-suodatintyökalu etsii toistuvia kaavoja käsiteltävästä tiedosta. Grok-työkalun avulla lokiviestin sisältö voidaan pilkkoa osiin ja jäsentää omiin kenttiin, mikä tehostaa tiedon tulkintaa ja sen hakemista. (Elasticsearch n.d.) Kansioon luotiin kaksi suodatinta. Yksi suodatin Linux-palvelimen syslog-lokiviestejä varten ja toinen pfSensen lokeja varten. Windows-käyttöjärjestelmien lokit indeksoidaan muuttamattomina.

Ensimmäinen suodatin luotiin nimeltä *02-syslog-filter.conf* (Kuva 16). Suodatin etsii syslog-lokiviestejä. Koska Linux-palvelimet sekä pfSense-reititin molemmat käyttävät syslog-lokiviestejä, on ne erotettava toisistaan. Tätä varten syslog-lähettäjistä etsitään lokilähettäjää, jolla on pfSense reitittimen IP-osoite. Kun pfSensen lähettämät lokiviestit tunnistetaan, lisätään sille pfSense- ja Ready-tunniste. Seuraavaksi suodatin etsii kaikki lokiviestit, joissa ei ole Ready-tunnistetta. Nämä leimataan syslog-tunnisteella. Lopuksi kaikista syslog-tyyppisistä lokiviesteistä poistetaan Ready-tunniste. pfSensen ja Linux-palvelimien lokiviestit voidaan nyt erottaa toisistaan tagin avulla.

```
filter {
    if [type] == "syslog" {
    if [host] =~ /192\.168\.1\.1/ {
        mutate {
            add_tag => [ "pfSense", "Ready" ]
        }
    }
    if "Ready" not in [tags] {
        mutate {
            add_tag => [ "syslog" ]
        }
    }
    }
}
filter {
    if [type] == "syslog" {
        mutate {
            remove_tag => "Ready"
        }
    }
}
```

Kuva 16. Syslog-suodattimen alkuosa (Kuvaruutukaappaus).

Tiedoston loppuosaan kirjoitettiin suodatin Linux-palvelimien syslog-viesteille (Kuva 17). Oletuksena koko lokiviestin sisältö kerrotaan message-kentässä. Tätä varten käytettiin grok-suodatinta palastelemaan kentän tieto pienempiin osiin. Message-kentästä etsitään aikaleima, lähettäjän nimi, lokiviestin kirjoittamisesta vastannut ohjelma, prosessi-id sekä viestisisältö. Nämä tiedot viedään omiin kenttiin. Lopuksi syslogin aikaleima muotoillaan yhtenäiseksi.

```
filter {
    if "syslog" in [tags] {
        grok {
            match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp}
            %{SYSLOGHOST:syslog_hostname}
            %{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?: $
            add_field => [ "received_at", "%{@timestamp}" ]
        }
        syslog_pri { }
        date {
            match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM dd HH:mm:ss" ]
        }
    }
}
```

Kuva 17. Syslog-suodattimen loppuosa (Kuvaruutukaappaus).

Seuraavaksi luotiin patterns-kansio, johon ladattiin Pisanon (2015) luomat grok-kaavat pfSenseä varten (Liite 1). Grok-kaavat toimivat *%{Syntaksi:merkitys}*-parametrilla. Syntaksin avulla luku 10 voidaan tunnistaa numeroksi, kun taas pisteiden erottama numerosarja voidaan tunnistaa IP-osoitteeksi. Parametrilla *%{NUMBER:duration} %{IP:client}* numero voidaan tunnistaa ajan kestoksi, kun taas IP-osoite voidaan merkata asiakaskoneeksi. (Elasticsearch n.d.)

Kansioon */etc/logstash* ladattiin Maxmind yrityksen luoma ilmainen geolokaatiotietokanta, jota voidaan käyttää IP-osoitteiden jäljitykseen. Tietokannan avulla IP-osoitteen voi jäljittää tietylle alueelle, mutta sen avulla ei voi tarkasti selvittää, mistä osoitteesta tai taloudesta tietty IP-osoite on lähtöisin. (Maxmind 2016.) Logstash lataa tietokannan käyttöönsä ohjelman uudelleen käynnistyksen yhteydessä.

Seuraavaksi luotiin suodatin pfSensen lokeja varten (Liite 2). Tiedostoksi nimettiin *04-pfsense-filter.conf*, jolloin Logstash siirtyy seuraavaksi tähän suodattimeen aiemman syslog-suodattimen jälkeen. Konfiguraatiotiedoston suodatin jäsentää lokiviestistä aikaleiman lisäksi lisäksi IP- ja protokollatiedot sekä muun lokiviestin sisällön. Näihin tietoihin kohdistetaan aiemmin ladatut grok-kaavat, jotka jäsentävät tietoa vielä pienempään muotoon tunnistamalla suodatetusta tiedosta arvoja ja osoittamalla niille merkityksen. Tällä tavalla saadaan yksityiskohtaisempia tietoja, kuten lähde- ja tulo-osoitteet, protokollat, lähde- ja kohdeportit sekä sen, miten palomuuri on reagoinut yhteydenottoyritykseen. Suodattimen lopussa otetaan käyttöön Maxmindin GeoIP-tietokanta, jonka avulla käyttäjän IP-osoitteen voi paikantaa tietokannan tarjoamien leveys- ja pituusasteen koordinaattien avulla. Tietokannan kehittäjä korostaa, että koordinaatit eivät osoita lähteen kotitalouteen, vaan näyttävät mahdollisen alueen, mistä IP-osoite on peräisin. (Maxmind, 2016.)

Luotiin vielä kolmas suodatin, jonka tarkoitus on poistaa ylimääräisiä tietokenttiä (Kuva 18). Esimerkiksi message-kentän tieto viedään omiin kenttiin, jolloin alkuperäistä kenttää ei enää tarvita. Kenttien poistaminen pitää tehdä aina muiden suodattimien lopuksi, muuten tietokenttiä saattaa kadota suodatinten välillä, mikä voi johtaa siihen, että lokitieto voi olla puutteellista.

```
filter {
        mutate {
        remove_field => [ "syslog_severity_code",
        "syslog_facility","message","offset","beat",
        "@version","syslog_facility_code","input_type","host" ]
        }
}
```

Kuva 18. Viimeisellä suodattimella poistetaan ylimääräiset kentät (Kuvaruutukaappaus).

Viimeisessä konfiguraatiotiedossa määritellään mihin Logstashin prosessoima tieto lähetetään (Kuva 19). Tässä tapauksessa lokitieto lähetetään Elasticsearchin indeksoitavaksi. pfSensen lokiviestit viedään *logstash*-nimiseen indeksiin, kun taas Beat-ohjelmien lokitieto viedään omaan indeksiin, jonka nimi koostuu päivämäärästä ja käytetyn Beat-ohjelman nimestä. Tämä tarkoittaa sitä, että kaikkien Linux-palvelimien lokitieto tallennetaan Filebeat-nimiseen indeksiin, kun taas Windows-palvelimien tiedot kerätään Winlogbeat-indeksiin.

```
output {
    if "pfSense" in [tags] {
        elasticsearch {
            hosts => ["localhost:9200"]
            index => "logstash-%{+YYYY.MM.dd}"
        }
    }
    else {
        elasticsearch {
            hosts => ["localhost:9200"]
            manage_template => false
            index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
            document_type => "%{[@metadata][type]}"
        }
    }
}
```

Kuva 19. Lähdöllä määritetään mihin tieto lähetetään (Kuvaruutukaappaus).

4.2.4 Kibanan konfigurointi

Kibanaa varten ladattiin Elasticin sivuilta ohjelman verkkohallintapaneeli. Ohjauspaneeli otettiin käyttöön puretun kansion sisältämän asennusscriptin avulla. Kibanan konfigurointitiedostoon jouduttiin tekemään muutoksia, jotta verkkokäyttöliittymään saatiin yhteys lähiverkon tietokoneiden kautta. Kibanan konfiguraatiotiedosto on nimeltään *kibana.yml* ja se sijaitsee polussa */etc/kibana/*.

Tiedostoon tehtiin osoiteohjaukseen liittyviä muutoksia. Ohjelman portti- ja IP-osoiteasetuksista poistettiin kommentointi, jotta niihin tehdyt muutokset saataisiin käyttöön. *Server.port*-asetukseen määritettiin, minkä portin kautta yhteys Kibanaan muodostetaan. Porttinumeroksi määritettiin 5601, joka on Kibanan oletusporttinumero. *Server.host*-asetuksella Kibana-palvelu sidotaan haluttuun IP-osoitteeseen. Tähän määritettiin Elastic-palvelimen lähiverkon IP-osoite. Näiden tietojen avulla lähiverkon tietokoneet voivat muodostaa yhteyden Kibanan verkkopohjaiseen käyttöliittymään. Tämän lisäksi tiedostoon määritettiin Elasticsearchin sijainti *elasticsearch.url*-asetuksella. Asetukseen kirjattiin *localhost*, sillä Elasticseach sijaitsee samalla palvelimella kuin Kibana.

4.3 **Asiakaskoneiden konfigurointi**

Tässä vaiheessa Logstash on konfiguroitu vastaanottamaan ja prosessoimaan saapuvia lokeja ja ohjaamaan niitä Elasticsearchin indeksoitavaksi. Kibana on asennettu ja valmiina lukemaan lokeja Elasticsearchilta. SSL-sertifikaatit on luotu ja palomuuriin on tehty tarvittavat porttiohjaukset palveluiden tietoliikenteen sallimiseen. Seuraavaksi asiakaskoneille asennetaan ja konfiguroidaan Beat-ohjelmat, jotka huolehtivat lokitiedon lähettämisestä. Beat-ohjelmat konfiguroimaan käyttämään SSL-suojausta lokien lähettämiseksi. Lokien lähettämiseen käytetään Filebeat- ja Winlogbeat-ohjelmia.

4.3.1 Lokien lähetys Linux-palvelimilta ja pfSense-reitittimeltä

Linux-asiakaskoneisiin asennetaan Filebeat-ohjelma, joka vastaa lokitiedon lähettämisestä Elastic-palvelimelle. Ubuntu-palvelimille haettiin ensin Elasticin PGP-avain, joka ladattiin apt-paketinhallintaan. Elastic-pakettivarasto lisättiin apt-paketinhallinnan latauslähteisiin, jonka jälkeen Filebeat-ohjelma haettiin ja asennettiin paketinhallinnan kautta. RedHat-pohjainen CentOS-palvelin käyttää Debian-pohjaisesta Ubuntusta poiketen rpm-paketinhallintaa. Elasticin PGP-avain ladattiin rpm-*työkalun import*-komennon avulla. Elastic-pakettivarasto luotiin nano-tekstinkäsittelyohjelmalla */etc/yum.repos.d/*-polkuun repo-loppupäätteellä. Elastic-pakettivarasto koostuu muun muassa pakettivaraston nimestä sekä pakettien- ja PGP-avaimen URL-osoitteesta. Tämän jälkeen Filebeat asennettiin CentOS-palvelimelle yum-työkalun avulla.

Filebeat on konfiguroitava ennen käyttöönottoa. Filebeatin konfiguraatiotiedosto löytyy polusta */etc/filebeat/filebeat.yml*. Kuvassa 20 nähdään konfiguraatiotiedoston asetukset. Konfiguraatiotiedostossa määritetään SSL-yhteys, lokitiedostot jotka halutaan siirrettävän, lähetettävien lokien tyyppi ja IP-osoite mihin lokit lähetetään. Tiedoston *prospectors*-kenttään määritetään lokityypit ja tiedostot, joiden tieto lähetetään Elastic-palvelimelle. Jokainen *prospector*-kentän loki merkataan erikseen väliviivalla, joka sijoitetaan lokipolun eteen. (Elasticsearch n.d.)

```
filebeat.prospectors:
- input_type: log
  document_type: syslog
  paths:
    - /var/log/auth.log
    - /var/log/syslog
  registry_file: /var/lib/filebeat/registry

output.logstash:
 hosts: ["192.168.1.16:5044"]
 ssl.certificate_authorities: ["/certificate/logstash.crt"]

shipper:

logging:
files:
rotateeverybytes: 10485760 # = 10MB
```

Kuva 20. Ubuntu-palvelimen Filebeat-asetukset (Kuvaruutukaappaus).

Oletuksena Filebeat lähettää kaikki */var/log*-kansiopolussa sijaitsevat lokit lokienkeräilijälle. Asetus poistettiin käytöstä ja lähetettäviksi lokitiedostoiksi määriteltiin Ubuntu-palvelimella *auth.log* ja *syslog*. *Auth.log*-tiedostoon järjestelmä kirjaa valtuutusta vaativat tapahtumat, kuten käyttäjäkirjautumiset, varmennukset ja *sudo*-komennolla suoritetut tehtävät. Ubuntu tallentaa järjestelmää sekä ohjelmia koskevat lokit *Syslog*-nimiseen tiedostoon (Wallen 2016). Ubuntusta poiketen CentOS-käyttöjärjestelmä tallentaa päivittäiset lokit omiin tekstitiedostoihin, jotka se nimeää *messages*-nimellä. Tietoturvalokit CentOS-järjestelmä tallentaa *secure*-nimisiin tiedostoihin. Nämä poikkeavuudet otettiin huomioon konfiguroitaessa Filebeat-ohjelmaa CentOS-palvelimelle.

pfSense-reitittimen graafisen käyttöliittymän lokiasetuksissa on sisäänrakennettu ominaisuus lokien lähetykselle, joka otetaan käyttöön. Palvelimet-kenttään kirjattiin Elastic-palvelimen osoite ja 5140 porttinumero, johon lokit lähetetään. Lokisisällöstä voidaan valita mitä lokeja lähetetään käsiteltäväksi. Valittiin kaikki vaihtoehdot. Näihin kuului muun muassa järjestelmän tapahtumat, palomuurilokit, oletusyhdyskäytävän tapahtumat sekä käyttäjäkirjautumiset.

4.3.2 Lokien lähetys Windows-palvelimilta

Windows-käyttöjärjestelmille on oma lokienlähetysohjelma nimeltä Winlogbeat, joka lähettää tapahtumienhallinnan lokitietoja eteenpäin. Ohjelman asennuspaketti löytyy Elasticin kotisivuilta. Ohjelma asennettiin powershell-scriptin avulla. Asennus-scripti muutti Winlogbeat-sovelluksen palveluksi. Kansiosta löytyi *Winlogbeat.yml*-konfiguraatiotiedosto. W*inlogbeat.event_logs*-rivin alle määritetään, mitkä lokit ja lokityypit kerätään eteenpäin lähetettäväksi.

Kerättäviksi tapahtumiksi valittiin tietoturvalokien lisäksi sovellus- ja järjestelmälokit sekä palvelimen toimialuepalveluihin liittyviä lokeja. Aluksi kerättiin myös tiedotuslokeja, mutta koska niiden määrä oli niin suuri, jätettiin ne käyttöön vain turvallisuuslokien kohdalla. Kuvassa 21 nähdään säännöt lokien keräykselle.

```
winlogbeat.event_logs:
  - name: Application
    ignore_older: 72h
    level: critical, error, warning
  - name: Security
    level: info, critical, error, warning
  - name: System
    level: critical, error, warning
  - name: DNS Server
    ignore_older: 24h
    level: critical, error, warning
  - name: Active Directory Web Services
    ignore_older: 24h
    level: critical, error, warning
  - name: Directory Service
    ignore_older: 24h
    level: critical, error, warning
```

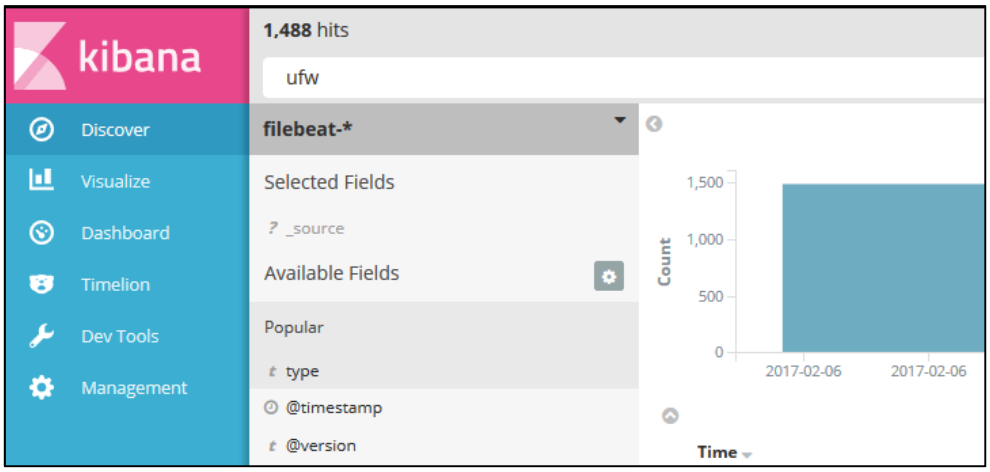
Kuva 21. Winlogbeatin asetukset lokien keräykseen (Kuvaruutukaappaus).

Ennen ohjelman käyttöä on määritettävä mihin lokit lähetetään sekä SSL-sertifikaatti ja avain yhteyden suojaamiseen, jos sellaisia on käytössä. Lokit määrättiin lähetettäväksi Elastic-palvelimelle lisäämällä *hosts*-kenttään palvelimen IP-osoite ja porttinumero. Elastic-palvelimen sertifikaatti siirrettiin Elastic-palvelimelta WinSCP-ohjelman avulla *Winlogbeat*-kansion alle. Asetuksissa osoitetaan sertifikaatin sijainti. Muutoksien jälkeen konfiguraatiotiedosto testattiin winlogbeat-ohjelman komennolla *.\winlogbeat.exe -c .\winlogbeat.yml -configtest -e*. Konfiguraatiotestin avulla kirjoitusvirheet ja ongelmat voidaan huomata ennen asetuksien käyttöönottoa. Lopuksi palvelu käynnistettiin powershell-komennolla *start-service winlogbeat*.

4.4 Asennuksen jälkeinen testaus

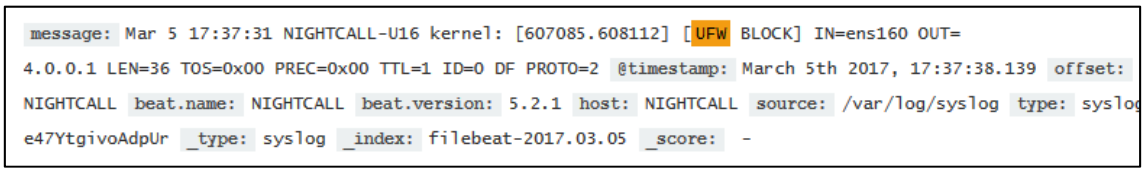
Kibanan verkkokäyttöliittymään pääsee Elastic-palvelimen IP-osoitteella ja porttinumerolla. Ensimmäisenä Kibanan verkkokäyttöliittymästä valitaan oletusindeksi. Indeksi haetaan nimen avulla. Aiemmassa luvussa (4.2.3) luotiin Logstashin lähtö-konfigurointi, jolla määriteltiin, mihin indeksiin Beat-ohjelmien ja pfSense-reitittimen lokitieto lähetetään. Haettiin Filebeat-indeksi ja valittiin se oletusindeksiksi syystä, että lokitietoa kerätään enimmäkseen Linux-palvelimilta. Lokit haetaan aikaleiman mukaisesti. Samalla lisättiin myös pfSensen ja Windows-palvelimen indeksit.

Kibanan käyttöliittymä on yksinkertainen ja helppokäyttöinen. Vasemmalta puolelta löytyy sivuston navigointipalkki. Elasticsearchin indeksoimaa lokitietoa voidaan hakea discover-sivun kautta. Hakukentän alta löytyy alasvetovalikko, josta valitaan minkä Beat-tyypin lokeja halutaan selata. Tämän hetkisten asetuksien mukaisesti lokeja voi selata ryhmittäin. Esimerkiksi Linux-palvelimien lokeja voidaan selata valitsemalla indeksivaihtoehdoista Filebeat (Kuva 22). Kaikkien indeksien tieto voidaan hakea, jos luodaan erillinen *-indeksi, joka kattaa kaikkien indeksien tulokset.



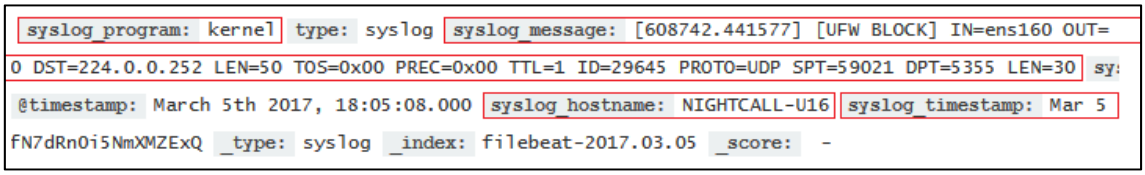
Kuva 22. Indeksin valinta Kibanan hakukentästä (Kuvaruutukaappaus).

Tässä vaiheessa lokeja alkoi ilmestyä nopeasti jokaisen uuden Beat-ohjelman asennuksen jälkeen. Kibana näyttää hakusivulla lokit aikajärjestyksessä uusimmasta vanhimpaan. Hakusivulla on myös pylväsdiagrammi, josta nähdään kerättyjen lokien määrä ajallisesti. Hakutuloksia voi rajata ajan mukaan. Elasticsearchin hakukone on joustava. Hakusanalla etsittäessä Elasticsearch käy läpi kaikki lokit ja näyttää lokit, joista kyseinen sana löytyy (Kuva 23). Tarkempien hakutulosten saamiseksi voidaan käyttää Lucenen hakusyntaksia tai JSON-pohjaista DSL-hakusyntaksia. Lucenen hakusyntaksi mahdollistaa AND-, OR- ja NOT-sanoista koostuvien Boolen-operaattoreiden käytön. Sen avulla haut voidaan kohdistaa myös eri kenttien tietoihin. Rajatun lukuarvotiedon hakemiseen voidaan käyttää sulkumerkkejä, joita voidaan hyödyntää esimerkiksi tietyntyyppisten virheviestien hakemisessa. (Elasticsearch n.d.)



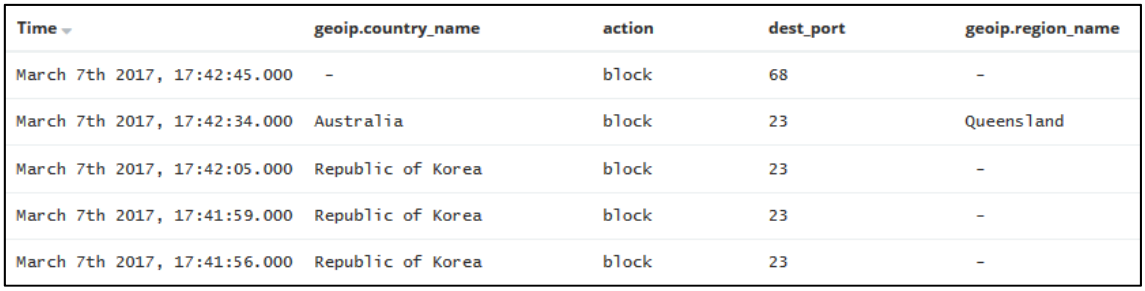
Kuva 23. Palomuurilokin hakutulos (Kuvaruutukaappaus).

Kuvassa 23 on otos palomuurilokista, jossa ei ole Logstashin suodatinta käytössä. Lokin *message*-kenttä sisältää viestin lisäksi aikaleiman, lähettäjän sekä ohjelman nimen, joka lokin loi. Vertailun vuoksi alla näkyvässä kuvassa 24 on lokiviesti, jonka prosessoinnissa käytettiin aiemmassa luvussa (4.2.3) kirjoitettua suodatinta. Lokiviestin *message*-kentän sisältö on purettu neljään eri kenttään viestin läpikäymisen helpottamiseksi.



Kuva 24. Palomuuriloki Logstashin suodatuksella (Kuvaruutukaappaus).

Oletuksena hakusivulla nähdään kooste, jossa nähdään kaikki lokista koottu tieto. Tiedon esitettävää määrää voidaan rajata lokitietokenttien avulla, jotka ovat valittavissa haun alapuolella. Tuloksena on rajattu lokitieto, jossa nähdään vain valitut kentät (Kuva 25).



Time	geoip.country_name	action	dest_port	geoip.region_name
March 7th 2017, 17:42:45.000	-	block	68	-
March 7th 2017, 17:42:34.000	Australia	block	23	Queensland
March 7th 2017, 17:42:05.000	Republic of Korea	block	23	-
March 7th 2017, 17:41:59.000	Republic of Korea	block	23	-
March 7th 2017, 17:41:56.000	Republic of Korea	block	23	-

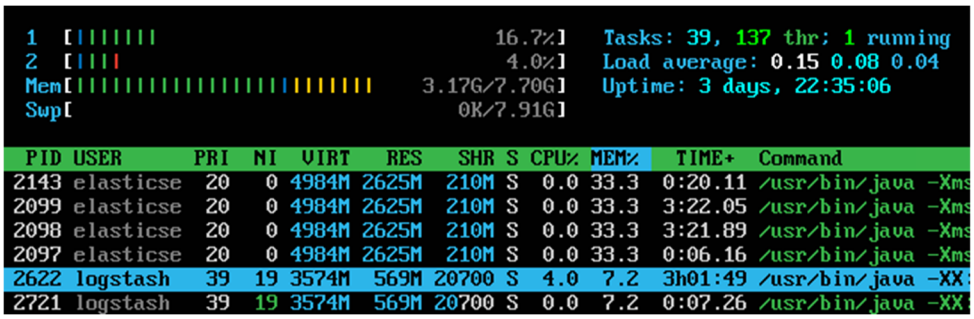
Kuva 25. Esitettävien lokikenttien rajaus (Kuvaruutukaappaus).

Hakukoneella haetut lokitiedot voidaan avata tarkisteltavaksi, joko taulukko- tai JSON-muodossa. Molemmissa lukutiloissa nähdään koko lokiviestin sisältö kaikkine kenttineen. Kuvassa 26 nähdään esimerkki indeksoidusta lokitiedosta. Lokin viestiosuus on purettu ja jaettu omiin kenttiin aiemmassa osassa kirjoitetun suodattimella avulla (4.2.3). Grok-työkalu erottaa alkuperäisestä viestistä viestin aikaleima, viestin lähettäjän ja sisällön omiin kenttiin. Alkuperäinen *message*-kenttä on poistettu ylimääräisenä.


```
{
  "_index": "filebeat-2017.03.05",
  "_type": "syslog",
  "_id": "AVqfN8ZVn0i5NmXMZExS",
  "_score": null,
  "_source": {
    "source": "/var/log/syslog",
    "syslog_program": "kernel",
    "type": "syslog",
    "syslog_message": "[1925804.174418] [UFW BLOCK] IN=e
PROTO=UDP SPT=59021 DPT=5355 LEN=30 ",
    "syslog_severity": "notice",
    "tags": [
      "beats_input_codec_plain_applied"
    ],
    "@timestamp": "2017-03-05T16:05:08.000Z",
    "syslog_hostname": "STARDUST",
    "syslog_timestamp": "Mar  5 18:05:08",
    "received_at": "2017-03-05T16:05:15.079Z"
  },
```


Kuva 26. Lokiviesti JSON-muodossa (Kuvaruutukaappaus).

Kibanan testauksen jälkeen tarkistettiin laitteistoresurssien kulutus htop-ohjelmalla (Kuva 27). Htop on ohjelma, jolla voidaan seurata laitteistoresurssien kulutusta reaaliajassa. Järjestelmän tilaa tarkkailtiin normaalissa tilassa, jolloin järjestelmään ei tehty muutoksia. Elastic-ohjelmat sekä järjestelmän ohjelmat käyttivät lähes puolet virtuaalikoneen muistikapasiteetista, joten 8 GB:n varaaminen virtuaalikoneelle oli jälkeenpäin ajateltuna hyvä ratkaisu. Prosessoritehon kulutus pysyi pienenä satunnaisia piikkejä huomioimatta. Toisen prosessoriytimen vähentäminen virtuaalikoneesta voi olla järkevä ratkaisu järjestelmän optimoinnin kannalta.




```
1  [|||||||                                16.7%]   Tasks: 39, 137 thr; 1 running
2  [||||                                     4.0%]   Load average: 0.15 0.08 0.04
Mem[||||||||||||||||||||||||||||||||||||||| 3.17G/7.70G]   Uptime: 3 days, 22:35:06
Swp[                                         0K/7.91G]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 2143 elasticse  20   0 4984M 2625M  210M S  0.0 33.3  0:20.11 /usr/bin/java -Xms
 2099 elasticse  20   0 4984M 2625M  210M S  0.0 33.3  3:22.05 /usr/bin/java -Xms
 2098 elasticse  20   0 4984M 2625M  210M S  0.0 33.3  3:21.89 /usr/bin/java -Xms
 2097 elasticse  20   0 4984M 2625M  210M S  0.0 33.3  0:06.16 /usr/bin/java -Xms
 2622 logstash   39  19 3574M  569M 20700 S  4.0  7.2  3h01:49 /usr/bin/java -XX:
 2721 logstash   39  19 3574M  569M 20700 S  0.0  7.2  0:07.26 /usr/bin/java -XX:
```


Kuva 27. Asennuksen jälkeinen laitteistoresurssien kulutus (Kuvaruutukaappaus).

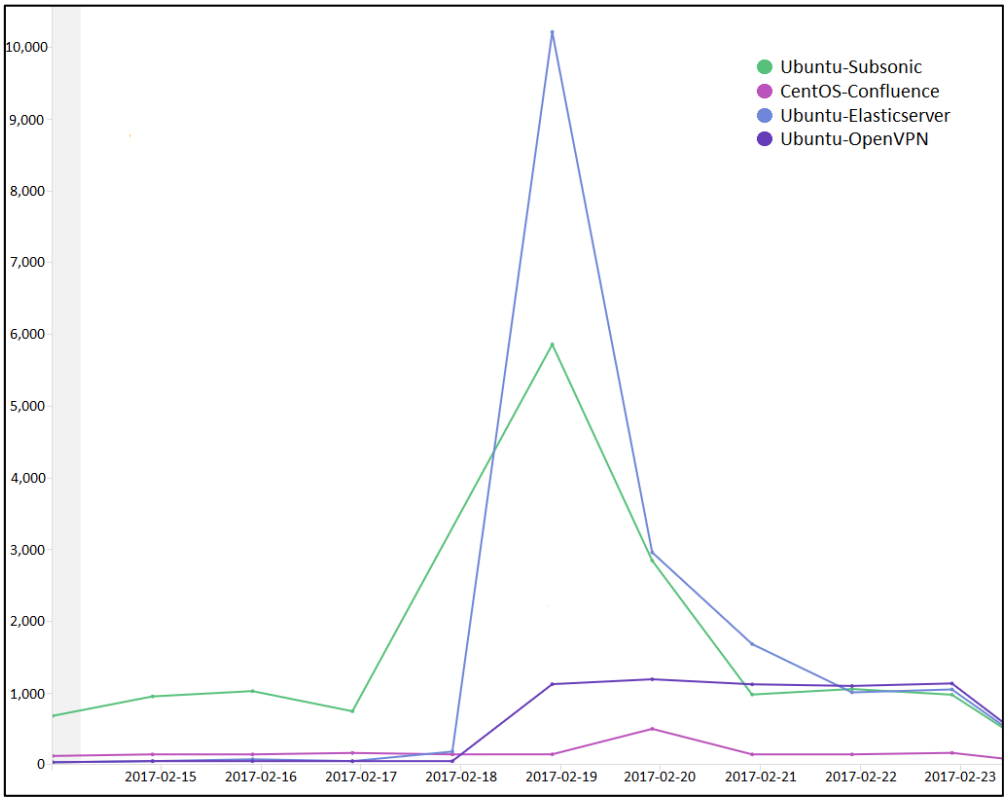
Kiintolevyn kapasiteetti tarkistettiin VMware virtualisointialustan ESXi-hallintapaneelista. Virtuaalikoneelle luotiin alussa 35 GB:n kokoinen kiintolevyosio, josta asennuksen jälkeen oli käytetty noin 21 GB. Vapaata tilaa osiolle jäi vielä 14 GB:n verran. Lokien osuus käytetystä tilasta oli 0,4 GB. Lukema saatiin käyttämällä *du -sh* -komentoa, jolla voidaan laskea kansio sisällön viemä kokonaistila. Lokit sijaitsivat */var/lib/elasticsearch/elasticsearch/nodes/0/indices*-polussa, johon komentoa käytettiin. Lokien kokonaiskoon perusteella päädyttiin siihen tulokseen, että tässä vaiheessa ei ole tarvetta luoda uutta kovalevyosiota lokeja varten.

4.5 **Tiedon visualisointi**

Kibana mahdollistaa tiedon visualisoinnin sisäänrakennettujen työkalujen avulla. Visualisoinneilla voidaan rakentaa hallintapaneeleja, joilla voidaan seurata eri lokitiedon keruuta reaaliajassa. Kibanassa visualisointi perustuu lokitiedosta saadun tiedon esittämiseen. Visualisoinnissa voidaan käyttää hakusanoja, ajanrajausta ja lokiviestistä jäsennettyjä tietokenttiä, joiden avulla kerätystä tiedosta voidaan muodostaa graafisia esityksiä. Esitykset voivat kerätä tuloksia reaaliajassa. Kibanassa on mahdollista luoda tallennetuista visualisoinneista kokonainen hallintapaneeli, johon visualisoinnit voidaan tuoda. Visualisoidusta tiedosta voidaan helpommin erottaa toistuvat kaavat tai poikkeamat.

Visualisointia luodessa on tiedettävä, mitä sen avulla halutaan selvittää. Visualize-sivulta valitaan ensin tiedon visualisointitapa. Visualisointitavoista voi valita muun muassa piirakkakaavion, viivakaavion, pylväskaavio, taulukon, avainsanapilven tai kartan. Seuraavaksi valitaan hakuindeksi, johon haku kohdistetaan. Esimerkiksi viivakaaviota tehdessä valitaan ensin tapa, jolla Y-akselin tieto lasketaan. Tiedon laskutapa voi olla esimerkiksi perinteisen yhteenlaskun tai keskiarvon mukaan. Sitten voidaan määrittää X-akseli, jolla tieto esitetään lineaarisesti kaaviossa. Tieto voidaan esittää esimerkiksi perinteisellä aikajanalla. Muita vaihtoehtoja ovat suodattimien, vaihteluvälien ja termien käyttäminen. X-akselin lisäksi tieto voidaan hajauttaa eri janoille esimerkiksi tietokoneiden nimien mukaan. Lopuksi hakukoneelle esitetään haku ja tuloksista kerätty tieto piirtyy kaaviolle.

Tiedon visualisointia testattiin Kibanan työkalujen avulla. Visualisoinnin avulla huomattiin, kuinka paljon palvelimilla tehdyt muutokset vaikuttivat generoitavaan lokimäärään. Elastic-palvelimen lokimäärät nousivat 55-kertaisiksi päivänä, jolloin palvelimelle tehtiin eniten muutoksia (Kuva 28). Kuvasta voidaan nähdä kolme edeltävää päivää, jolloin Linux-palvelimien lokien generointi oli huomattavasti vähäisempää. Lokimäärän kasvu havaittiin myös muilla palvelimilla päivinä, jolloin palvelimille tehtiin muutoksia lokien keskittämiseksi. Visualisointi osoittaa kuinka helposti järjestelmiin tehtävät muutokset voivat näkyä lokimäärän lisääntymisenä. Tällä tavoin järjestelmien poikkeukset ovat helpommin huomattavissa, ja poikkeuksien syitä voidaan aloittaa etsimään.

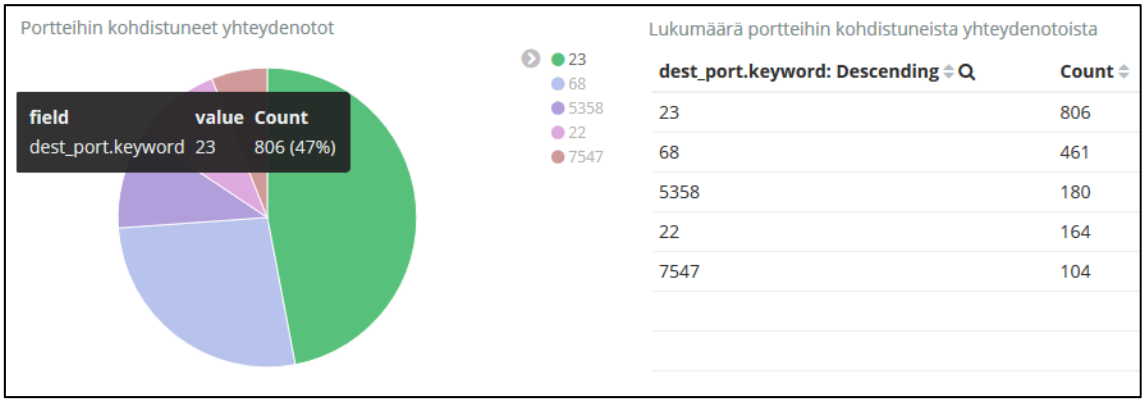


Kuva 28. Linux-palvelimien lokimäärät päivinä, jolloin järjestelmiä konfiguroitiin ja asennettiin (Kuvaruutukaappaus).

pfSensen palomuurilokitiedoista luotiin useita visualisointeja. Ensin luotiin lämpökarttapohja, johon tuotiin kaikesta kerätystä lokitiedosta GeoIP-paikannustiedot, *geoip.location*-tunnisteen avulla, joka sisältää Maxmindin tietokannasta saadut leveys- ja pituusasteen koordinaatit. Paikannustiedot tuotiin yhteen, jolloin kartalta voidaan nähdä, mistä päin maailmaa yhteyttä on yritetty muodostaa. Luotiin myös taulukkolistaus, jossa lasketaan yhteen eri maista tulleet yhteydenottoyritykset *geoip.country_name*-tunnisteen avulla. Kolmantena visualisointina luotiin yksinkertainen laskin, joka laskee yhteen esitettyjä hakutuloksia. Hakutuloksien aikarajaksi valittiin päivän sisällä kertyneet tulokset. Hakusanaksi kirjattiin *action:block*, jolla etsitään palomuurin estämät yhteydenotot. Luotiin myös pylväsdiagrammi päivittäisistä palomuuriestoista sekä piirakkakaavio, joka laskee ja näyttää *dest_port*-tunnisteen avulla viisi porttinumeroa, joihin yhteydenottoja on eniten yritetty tehdä. Palomuurilokien visualisoinnit tallennettiin ja vietiin hallintapaneelinäkymään. (Liite 3.)

Palomuurilokien visualisointi auttoi järjestelmän ylläpitäjää hahmottamaan palomuurin kohtaamia verkon ulkopuolisia uhkia. Sen avulla huomattiin, että 47 % päivän aikana saaduista yhteydenotoista kohdistui porttiin 23 (Kuva 29). Portti on Telnet-protokollan yhteysportti, jonka avulla voidaan luoda etäyhteys verkossa olevaan tietokoneeseen (Indiana University, 2016). 10 % päivän aikana kerätyistä osumista kohdistui port-

tinumeroon 22. Kyseistä porttinumeroa käytetään SSH-etäyhteyden muodostamiseen ja voi olla merkki siitä, että joku on yrittänyt selvittää, onko IP-osoitteen päässä tietokonetta, johon voisi kirjautua etänä. Portin liikenne oli sallittu tietylle IP-osoitteelle, mutta näiden tietojen perusteella päätettiin varmuuden vuoksi ottaa portti toistaiseksi kokonaan pois käytöstä siksi aikaa, kun portti saadaan paremmin suojattua.



Kuva 29. Päivän aikana kertyneet palomuuriestot (Kuvaruutukaappaus).

Lokihallintajärjestelmän avulla palvelimien lokit ovat nyt yhtenäisiä ja järjestelmien vikalokit ovat haettavissa verkkohallintapaneelista, joka helpottaa etenkin järjestelmien välisten konfliktien selvitystyötä. Lokihistorian avulla nykyisiä tapahtumia voidaan verrata aiempiin lokitietoihin, jolloin poikkeavuuksien löytäminen on huomattavasti helpompaa Kibanan visualisointeja hyödyntämällä.

5 YHTEENVETO

Halusin haastaa opinnäytetyössäni itseni valitsemalla aiheekseni minulle ennalta täysin tuntemattoman ja aluksi monimutkaiselta vaikuttavan järjestelmän käyttöönoton. Halusin heittää itseni tulikokeeseen ja katsoa miten selviytyisin siitä. Koin, että opinnäytetyö olisi erinomainen tilaisuus kehittää osaamistani oppimalla uutta ja mielestäni onnistuin siinä.

Halusin ensin syventyä lokeihin ja kerätä niistä tietoa kaventaakseni ymmärrystä aiheesta, jotta ymmärtäisin asiaan liittyviä konsepteja käytännön osuuteen astuttaessa. Yllätyin positiivisesti, kuinka paljon aiheestani löytyi tietoa ja monesta eri näkökulmasta. Esittämiini tutkimuskysymyksiini sain työn teoriaosuudessa vastaukset. Teoriaosuudesta sain myös paljon hyvää tietoa, jota käytin hyödykseni lokijärjestelmän käyttöönoton suunnittelussa. Opin lokiprotokollista ja niiden tietoturvaheikkouksista, jonka pohjalta päädyin käytännön osuudessa hyödyntämään SSL-sertifikaatteja asiakaskoneilta lähetettyjen lokien suojaamiseksi.

Tutustuin Elastic-lokihallintajärjestelmään kuuluvien ohjelmien toimintatapoihin. Opinnäytetyön käytännön osuudessa pääsin asentamaan Elastic-lokihallintajärjestelmän, joka koostui Elasticsearchista, Logstashista ja Kibanasta. Muutamaan otteeseen sain oppia vaikeimman kautta asioista, joista ei ollut mainintaa asennusoppaissa. Opin miten Logstashin voi konfiguroida suodattamaan ja jäsentämään lokitietoa. Elastic-lokihallintajärjestelmä osoittautui hyvin tehokkaaksi työkaluksi lokien keräykseen, sen muuttamiseen samaan formaattiin, indeksointiin ja talletukseen.

Perehdyin Kibanan visualisointityökaluihin ja niiden hyödyntämiseen lokitiedon visualisoinnissa. Visualisoinneilla pystyin osoittamaan, kuinka järjestelmien konfiguraatiot ja ohjelmien asennukset voivat näkyä huomattavina piikkeinä järjestelmien generoimassa lokimäärässä. Järjestelmien poikkeukset on jatkossa helpompi huomata lokitiedosta koostuvien visualisointien avulla. Visualisointien luonnissa käytin hyväkseni myös pfSensen palomuurin lokitietoja, joiden avulla loin useita visualisointeja. Tallennetut pfSense-visualisoinnit vein yhteiseen hallintapaneeliin. Hallintapaneelin osoittama tieto auttoi minua paremmin hahmottamaan palomuurin kohtaamia verkon ulkopuolisia uhkia. Visualisoinneilla minulle selvisi, mitkä ylläpitämäni palomuurin portit ovat suurimpia tietoturvauhkia. Tämän tiedon perusteella poistin käytöstä SSH-yhteydestä vastaavan portin siksi aikaa, kun portti saadaan paremmin suojattua.

Onnistuin tavoitteissani mielestäni hyvin. Tuloksena oli toimiva lokihallintajärjestelmä, joka kerää toimialueen palvelimilta ja pfSense-reitittimeltä lokitietoa käyttäjän analysoitavaksi. Elastic-lokihallintajärjestelmän kehitys tulee jatkossa jatkumaan. Ajatuksena olisi suojata pfSensen-lokiyhteys ja siirtää lokit pois palvelimen SSD-järjestelmäosiosta perinteiseen kovalevyosioon. Lokitiedon seurantaa tulen aktiivisesti jatkamaan.

LÄHDELUETTELO

Aalto-Setälä, M. (2016). EU:n tietosuoja-asetus tulee – valmistaudu ajoissa. Haettu 5.3.2017 osoitteesta http://kauppaka-mari.fi/2016/03/31/eun-tietosuoja-asetus-tulee-valmistaudu-ajoissa/

Alapati, S. (2016). *Modern Linux Administration*. Ennakkojulkaisu. Sebastopol: O`Reilly Media.

Balabit (2017). The syslog-ng Open Source Edition 3.9 Administrator Guide – Introduction to syslog-ng. Haettu 29.1.2017 osoitteesta https://www.balabit.com/sites/default/files/documents/syslog-ng-ose-latest-guides/en/syslog-ng-ose-guide-admin/html/ch01s01.html

Charter, B. (2008). EVTX and Windows Event Logging. Haettu 31.1.2017 osoitteesta https://www.sans.org/reading-room/whitepapers/logging/evtx-windows-event-logging-32949

Chhajed, S. (2015). *Learning ELK Stack*. Birmingham: Packt Publishing Ltd.

Chuvakin, A., Schmidt, K. & Phillips, C. (2012). *Logging and Log Management*. Waltham: Syngress / Elsevier.

Cisco (n.d.). Configuring SNMP Support. Haettu 3.2.2017 osoitteesta http://www.cisco.com/c/en/us/td/docs/ios/12_2/configfun/configuration/guide/ffun_c/fcf014.html#wp1017597

Covington, R. (2015). Forewarned is forearmed: Using log management to prevent data breaches. Haettu 4.3.2017 osoitteesta http://www.computerworld.com/article/3003123/security/forewarned-is-forearmed-using-log-management-to-prevent-data-breaches.html

Stenberg, D. (n.d.). Curl man page. Haettu 16.2.2017 osoitteesta https://curl.haxx.se/docs/manpage.html

Elasticsearch (n.d.). Elastic Stack [5.2] – Installing the Elastic Stack. Haettu 15.2.2017 osoitteesta https://www.elastic.co/guide/en/elastic-stack/current/installing-elastic-stack.html#install-order-elastic-stack

Elasticsearch (n.d.). Elasticsearch Reference [5.2] – Basic Concepts. Haettu 1.3.2017 osoitteesta https://www.elastic.co/guide/en/elasticsearch/reference/current/_basic_concepts.html

Elasticsearch (n.d.). Elasticsearch Reference [5.2] – Important Elasticsearch configuration. Haettu 1.3.2017 osoitteesta https://www.elastic.co/guide/en/elasticsearch/reference/current/important-settings.html

Elasticsearch (n.d.). Filebeat Reference [1.2] – Loading the Index Template in Elasticsearch. Haettu 16.2.2017 osoitteesta https://www.elastic.co/guide/en/beats/filebeat/1.2/filebeat-template.html

Elasticsearch (n.d.). Filebeat Reference [5.2] – Filebeat Prospectors Configuration. Haettu 17.2.2017 osoitteesta https://www.elastic.co/guide/en/beats/filebeat/current/configuration-filebeat-options.html

Elasticsearch (n.d.). Filebeat Reference [5.2] – How Filebeat Works. Haettu 17.2.2017 osoitteesta https://www.elastic.co/guide/en/beats/filebeat/current/how-filebeat-works.html

Elasticsearch (n.d.). Kibana User Guide [5.2] – Searching Your Data. Haettu 22.2.2017 osoitteesta https://www.elastic.co/guide/en/kibana/current/search.html

Elasticsearch (n.d.). Logstash Reference [5.2] – grok. Haettu 4.3.2017 osoitteesta https://www.elastic.co/guide/en/logstash/current/plugins-filters-grok.html

Elasticsearch (n.d.). Logstash Reference [5.2] – Stashing Your First Event. Haettu 18.2.2017 osoitteesta https://www.elastic.co/guide/en/logstash/current/first-event.html

Elasticsearch (n.d.). Products – Beats. Haettu 17.2.2017 osoitteesta https://www.elastic.co/products/beats

Elasticsearch (n.d.). Winlogbeat Reference [5.2] – Overview. Haettu 17.2.2017 osoitteesta https://www.elastic.co/guide/en/beats/winlogbeat/current/index.html

Elijah, P. (2015). Monitoring pfSense logs using ELK (ElasticSearch 1.7, Logstash 1.5, Kibana 4.1). Haettu 5.3.2017 osoitteesta https://elijah-paul.co.uk/updated-monitoring-pfsense-logs-using-elk-elasticsearch-logstash-kibana-part-1/

F-Secure (n.d). Tracking cookie. Haettu 26.2.2017 osoitteesta https://www.f-secure.com/sw-desc/tracking_cookie.shtml

Gormley, C. & Tong, Z. (2015). *Elasticsearch: The Definitive Guide*. Sebastopol: O`Reilly Media.

Grimes, R. (2012). Log Analysis Deep Dive. Haettu 25.1.2017 osoitteesta https://www.scribd.com/document/247819299/Loganalysis-Deep-Dive

Henkilötietolaki 1999/523. Haettu 5.3.2017 osoitteesta http://www.finlex.fi/fi/laki/ajantasa/1999/19990523

Hoffman, C. (2015). How to use PPAs to install bleeding-edge software in Ubuntu and Linux Mint. Haettu 25.2.2017 osoitteesta http://www.pcworld.com/article/2942171/how-to-use-ppas-to-install-bleeding-edge-software-in-ubuntu-and-linux-mint.html

IBM (2017). An overview of the SSL or TLS handshake. Haettu 16.2.2017 osoitteesta https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q009930_.htm

Indiana University (2016). What is telnet? Haettu 7.3.2017 osoitteesta https://kb.iu.edu/d/aayd

Kasvi, J. (2014). Verkkopalvelut seuraavat ja profiloivat käyttäjiään – myyvät tiedot mainostajille. Haettu 25.2.2017 osoitteesta http://yle.fi/uutiset/3-7354145

Kenneth, E. (2003). A Security Analysis of System Event Logging with Syslog. Haettu 29.1.2017 osoitteesta https://www.sans.org/reading-room/whitepapers/logging/security-analysis-system-event-logging-Syslog-1101

Kent, K. & Souppaya, M. (2006). Guide to Computer Security Log Management. Haettu 25.1.2017 osoitteesta http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-92.pdf

Kosola, L. (2016). Facebook säilöö jopa puhelutietoja – näin meistä rakennetaan profiileja mainostajille. Haettu 5.3.2017 osoitteesta http://yle.fi/aihe/artikkeli/2016/10/11/facebook-sailoo-jopa-puhelutietoja-nain-meista-rakennetaan-profiileja

Mauro, D. & Schmidt, K. (2005). *Essential SNMP 2nd Edition*. Sebastopol: O`Reilly Media.

Maxmind (2016). GeoIP2 Release Notes. Haettu 6.3.2017 osoitteesta https://dev.maxmind.com/geoip/geoip2/release-notes/

Messier, R. (2015). *Operating System Forensics*. Waltham: Syngress / Elsevier.

Microsoft (2015). Configure Computers to Forward and Collect Events. Haettu 2.2.2017 osoitteesta https://technet.microsoft.com/en-us/library/cc748890(v=ws.11).aspx

Microsoft (n.d.). Event Properties. Haettu 2.2.2017 osoitteesta https://technet.microsoft.com/en-us/library/cc765981(v=ws.11).aspx

Männikkö, P. (2008). Tietosuoja: Loki jättää jäljen. Haettu 3.2.2017 osoitteesta https://www.tietosuoja-lehti.fi/index.php?mid=2&pid=32&aid=2587

Oracle (2014). Performing TCP and UDP Administration With the netcat Utility. Haettu 1.3.2017 osoitteesta https://docs.oracle.com/cd/E36784_01/html/E37476/gnqeb.html

RFC 5424 (2009). The Syslog Protocol. IETF Trust. Haettu 28.1.2017 osoitteesta https://tools.ietf.org/html/rfc5424

Turnbull, J. (2016). *The Logstash Book*. Brooklyn: James Turnbull.

Ubuntu (n.d.). OpenSSL. Haettu 16.2.2017 osoitteesta https://help.ubuntu.com/community/OpenSSL

Valtiovarainministeriö (2009). Lokiohje. Haettu 5.2.2017 osoitteesta https://www.vahtiohje.fi/c/document_library/get_file?uuid=775179cb-6c54-4dfb-b65d-e925d47c61d2&groupId=10229

Varghese, L. (2013). Auditing File Access on File Servers. Blogijulkaisu 26.8.2016. Haettu 29.1.2017 osoitteesta https://blogs.technet.microsoft.com/mspfe/2013/08/26/auditing-file-access-on-file-servers/

Verizon (2015). PCI Compliance Report 2015. Haettu 5.3.2017 osoitteesta http://www.verizonenterprise.com/resources/report/rp_pci-report-2015_en_xg.pdf

Verizon (2016). Data Breach Investigations Report. Haettu 25.2.2017 osoitteesta http://www.verizonenterprise.com/resources/reports/rp_DBIR_2016_Report_en_xg.pdf

Viestintävirasto (2016). Lokien keräys ja käyttö. Ohje lokitietojen tallentamiseen ja hyödyntämiseen. Haettu 26.1.2017 osoitteesta https://www.viestintavirasto.fi/attachments/tietoturva/Lokitusohje.pdf

Wallen, J. (2016). Viewing Linux Logs from the Command Line. Haettu 17.2.2017 osoitteesta https://www.linux.com/learn/sysadmin/viewing-linux-logs-command-line

Warma, E. & Luomala, A. (2016). Miten valmistautua EU:n uuden tietosuoja-asetuksen vaatimuksiin? Blogijulkaisu 2.3.2016. Haettu 5.3.2017 osoitteesta http://www.castren.fi/fi/blogijauutiset/blogi-2016/miten-valmistautua-eun-uuden-tietosuoja-asetuksen-vaatimuksiin/

Liite 1 (1/3)

PFSENSEN GROK-KAAVAT

```
# Created 27 Jan 2015 by J. Pisano (Handles TCP, UDP, and ICMP log entries)
# Edited 14 Feb 2015 by Elijah Paul elijah.paul@gmail.com
# Edited 10 Mar 2015 by Bernd Zeimetz <bernd@bzed.de>
# taken from https://gist.github.com/elijahpaul/f5f32d4e914dcb7fedd2
# - adding PFSENSE_ prefix
# - adding carp patterns
#
# Usage: Use with following GROK match pattern
#%{PFSENSE_LOG_DATA}%{PFSENSE_IP_SPECIFIC_DATA}%{PFSENSE_IP_DATA}%{PFSE
#NSE_PROTOCOL_DATA}

PFSENSE_LOG_DATA
(%{INT:rule}),(%{INT:sub_rule}),,(%{INT:tracker}),(%{WORD:iface}),(%{WORD:reason}),(
%{WORD:action}),(%{WORD:direction}),(%{INT:ip_ver}),

PFSENSE_IP_SPECIFIC_DATA
(%{PFSENSE_IPv4_SPECIFIC_DATA}|%{PFSENSE_IPv6_SPECIFIC_DATA})

PFSENSE_IPv4_SPECIFIC_DATA
(%{BASE16NUM:tos}),,(%{INT:ttl}),(%{INT:id}),(%{INT:offset}),(%{WORD:flags}),(%{INT:p
roto_id}),(%{WORD:proto}),

PFSENSE_IPv4_SPECIFIC_DATA_ECN
(%{BASE16NUM:tos}),(%{INT:ecn}),(%{INT:ttl}),(%{INT:id}),(%{INT:offset}),(%{WORD:fla
gs}),(%{INT:proto_id}),(%{WORD:proto}),

PFSENSE_IPv6_SPECIFIC_DATA
(%{BASE16NUM:class}),(%{DATA:flow_label}),(%{INT:hop_limit}),(%{WORD:proto}),(%{I
NT:proto_id}),

PFSENSE_IP_DATA (%{INT:length}),(%{IP:src_ip}),(%{IP:dest_ip}),

PFSENSE_PROTOCOL_DATA
(%{PFSENSE_TCP_DATA}|%{PFSENSE_UDP_DATA}|%{PFSENSE_ICMP_DATA}|%{PFSEN
SE_CARP_DATA})

PFSENSE_TCP_DATA
(%{INT:src_port}),(%{INT:dest_port}),(%{INT:data_length}),(%{WORD:tcp_flags}),
(%{INT:sequence_number}),(%{INT:ack_number}),(%{INT:tcp_window}),(%{DATA:urg_
data}),(%{DATA:tcp_options})

PFSENSE_UDP_DATA (%{INT:src_port}),(%{INT:dest_port}),(%{INT:data_length})

PFSENSE_ICMP_DATA (%{PFSENSE_ICMP_TYPE}%{PFSENSE_ICMP_RESPONSE})
```

```
PFSENSE_ICMP_TYPE (?<icmp_type>(request|reply|unreachproto|unreachport|un-
reach|timeexceed|paramprob|redirect|maskreply|needfrag|tstamp|tstampreply)),

PFSENSE_ICMP_RESPONSE
(%{PFSENSE_ICMP_ECHO_REQ_REPLY}|%{PFSENSE_ICMP_UNREACHPORT}|
%{PFSENSE_ICMP_UNREACHPROTO}|%{PFSENSE_ICMP_UNREACHABLE}|%{PFSENSE_I
CMP_NEED_FLAG}|%{PFSENSE_ICMP_TSTAMP}|%{PFSENSE_ICMP_TSTAMP_REPLY})

PFSENSE_ICMP_ECHO_REQ_REPLY
(%{INT:icmp_echo_id}),(%{INT:icmp_echo_sequence})

PFSENSE_ICMP_UNREACHPORT
(%{IP:icmp_unreachport_dest_ip}),(%{WORD:icmp_unreachport_protocol}),(%{INT:icm
p_unreachport_port})

PFSENSE_ICMP_UNREACHPROTO
(%{IP:icmp_unreach_dest_ip}),(%{WORD:icmp_unreachproto_protocol})

PFSENSE_ICMP_UNREACHABLE (%{GREEDYDATA:icmp_unreachable})

PFSENSE_ICMP_NEED_FLAG (%{IP:icmp_need_flag_ip}),(%{INT:icmp_need_flag_mtu})

PFSENSE_ICMP_TSTAMP (%{INT:icmp_tstamp_id}),(%{INT:icmp_tstamp_sequence})

PFSENSE_ICMP_TSTAMP_REPLY
(%{INT:icmp_tstamp_reply_id}),(%{INT:icmp_tstamp_reply_sequence}),(%{INT:icmp_ts
tamp_reply_otime}),
(%{INT:icmp_tstamp_reply_rtime}),(%{INT:icmp_tstamp_reply_ttime})

PFSENSE_CARP_DATA
(%{WORD:carp_type}),(%{INT:carp_ttl}),(%{INT:carp_vhid}),(%{INT:carp_version}),(%{I
NT:carp_advbase}),(%{INT:carp_advskew})

DHCPD
(%{DHCPDISCOVER}|%{DHCPOFFER}|%{DHCPREQUEST}|%{DHCPACK}|%{DHCPINFOR
M}|%{DHCPRELEASE})

DHCPDISCOVER %{WORD:dhcp_action} from
%{COMMONMAC:dhcp_client_mac}%{SPACE}(\(%{GREEDYDATA:dhcp_client_hostnam
e}\))? via (?<dhcp_client_vlan>[0-9a-z_]*)(: %{GREEDYDATA:dhcp_load_balance})?

DHCPOFFER %{WORD:dhcp_action} on %{IPV4:dhcp_client_ip} to
%{COMMONMAC:dhcp_client_mac}%{SPACE}(\(%{GREEDYDATA:dhcp_client_hostnam
e}\))? via (?<dhcp_client_vlan>[0-9a-z_]*)
```

DHCPREQUEST %{WORD:dhcp_action} for %{IPV4:dhcp_client_ip}%{SPACE}(\(%{IPV4:dhcp_ip_unknown}\))? from %{COMMONMAC:dhcp_client_mac}%{SPACE}(\(%{GREEDYDATA:dhcp_client_hostname}\))? via (?<dhcp_client_vlan>[0-9a-z_]*)(: %{GREEDYDATA:dhcp_request_message})?

DHCPACK %{WORD:dhcp_action} on %{IPV4:dhcp_client_ip} to %{COMMONMAC:dhcp_client_mac}%{SPACE}(\(%{GREEDYDATA:dhcp_client_hostname}\))? via (?<dhcp_client_vlan>[0-9a-z_]*)

DHCPINFORM %{WORD:dhcp_action} from %{IPV4:dhcp_client_ip} via %(?<dhcp_client_vlan>[0-9a-z_]*)

DHCPRELEASE %{WORD:dhcp_action} of %{IPV4:dhcp_client_ip} from %{COMMONMAC:dhcp_client_mac}%{SPACE}(\(%{GREEDYDATA:dhcp_client_hostname}\))? via (?<dhcp_client_vlan>[0-9a-z_]*)

Liite 2

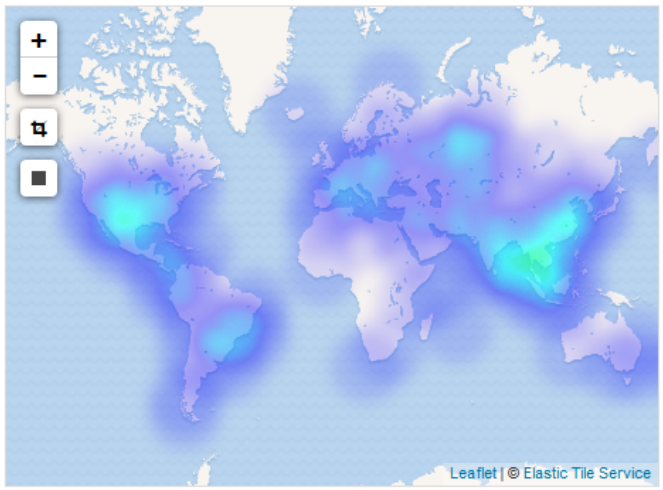
LOGSTASHIN PFSENSE-SUODATIN

```
filter {
  if "pfSense" in [tags] {
    grok {
      add_tag => [ "firewall" ]
        match => [ "message", "<(?<ev-
tid>.*)>(?<datetime>(?:Jan(?:uary)?|Feb(?:ruary)?|Mar(?:ch)?|Apr(?:il)?|May|Jun(?:e
)?|Jul(?:y)?|Aug(?:ust)?|Sep(?:tember)?|Oct(?:ober)?|Nov(?:ember)?|Dec(?:em-
ber)?)\s+(?:(?:0[1-9])|(?:[12][0-9])|(?:3[01])|[1-9]) (?:2[0123]|[01]?[0-9]):(?:[0-5][0-
9]):(?:[0-5][0-9])) (?<prog>.*?): (?<msg>.*)" ]
    }
    mutate {
      gsub => ["datetime","  "," "]
    }
    date {
      match => [ "datetime", "MMM dd HH:mm:ss" ]
      timezone => "Europe/Helsinki"
    }
    mutate {
      replace => [ "message", "%{msg}" ]
    }
    mutate {
      remove_field => [ "msg", "datetime" ]
    }
}
if [prog] =~ /^filterlog$/ {
    mutate {
      remove_field => [ "msg", "datetime" ]
    }
    grok {
      patterns_dir => "/etc/logstash/conf.d/patterns"
      match => [ "message",
"%{PFSENSE_LOG_DATA}%{PFSENSE_IP_SPECIFIC_DATA}%{PFSENSE_IP_DATA}%{PFSE
NSE_PROTOCOL_DATA}", "message",
%{PFSENSE_LOG_DATA}%{PFSENSE_IPv4_SPECIFIC_DATA_ECN}%{PFSENSE_IP_DATA}%
{PFSENSE_PROTOCOL_DATA}" ]
    }
    mutate {
      lowercase => [ 'proto' ]
    }
    geoip {
      add_tag => [ "GeoIP" ]
      source => "src_ip"
      database => "/etc/logstash/GeoLite2-City.mmdb"
    }
  }}
```

Liite 3

PFSENSEN PALOMUURILOKITIEDOISTA KOOSTUVA HALLINTAPANEELI

Lämpökartta estetyistä yhteyksistä



Palomuuriestot maista

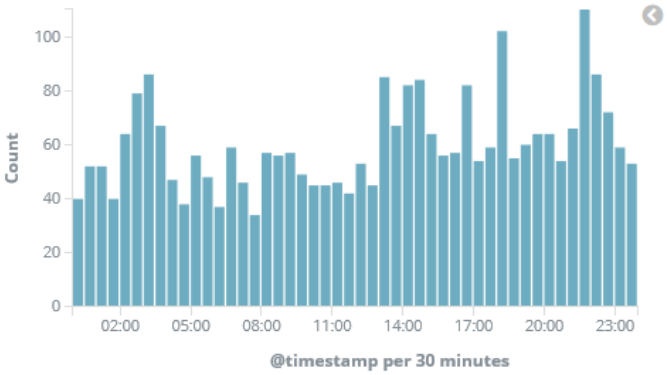
geoip.country_name.keyword: Descending	Count
China	625
United States	187
Russia	127
France	124
Vietnam	123
Republic of Korea	109
Taiwan	91

Päivittäiset estot

2,875

Count

Päivittäin estetyt yhteydet



Portteihin kohdistuneet yhteydenotot